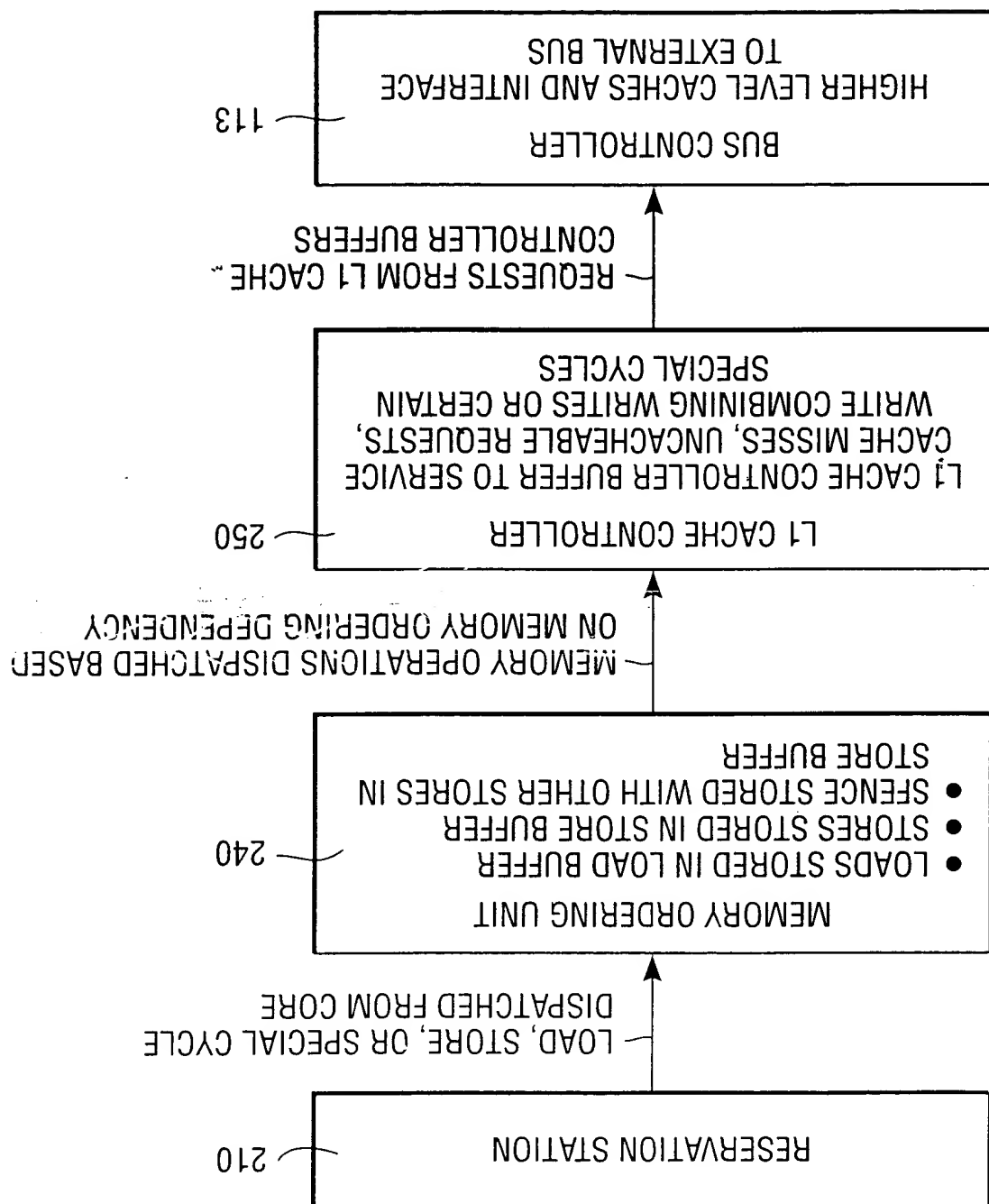


	L #	Hits	Search Text	DBs
1	L1	1130	(fenc\$3 synch\$ barrier) near10 load near10 (instruction operation)	USPAT; US-PGPUB
2	L2	330	(glob\$4 coheren\$2 consisten\$2) and 1	USPAT; US-PGPUB
3	L3	14	(glob\$4 coheren\$2 consisten\$2) near99 1	USPAT; US-PGPUB
4	L4	311	(post pre) and 1	USPAT; US-PGPUB
5	L5	4	(post pre) near99 1	USPAT; US-PGPUB
6	L6	143	2 and 4	USPAT; US-PGPUB
7	L7	178	2 not (6 3 5)	USPAT; US-PGPUB



	Docum ent ID	U	Title	Current OR
1	US 20030 20598 9 A1	<input type="checkbox"/>	Wound field synchronous machine control system and method	322/28
2	US 20030 16738 7 A1	<input checked="" type="checkbox"/>	Vector transfer system generating address error exception when vector to be transferred does not start and end on same memory page	712/4
3	US 20030 15423 8 A1	<input checked="" type="checkbox"/>	Peer to peer enterprise storage system with lexical recovery sub-system	709/201
4	US 20030 13571 7 A1	<input checked="" type="checkbox"/>	Method and apparatus for transferring vector data	712/222
5	US 20030 10638 3 A1	<input checked="" type="checkbox"/>	System having a unit for controlling a transmission line	74/336R
6	US 20030 09491 7 A1	<input checked="" type="checkbox"/>	WOUND FIELD SYNCHRONOUS MACHINE CONTROL SYSTEM AND METHOD	318/700
7	US 20030 09436 3 A1	<input checked="" type="checkbox"/>	Grounding system for rotating fixtures in electrically conductive mediums	204/198
8	US 20030 08425 9 A1	<input checked="" type="checkbox"/>	MFENCE and LFENCE micro-architectural implementation method and system	711/163
9	US 20030 07906 5 A1	<input checked="" type="checkbox"/>	Methods and apparatus for providing data transfer control	710/22
10	US 20030 07408 8 A1	<input checked="" type="checkbox"/>	Method and apparatus for providing distributed scene programming of a home automation and control system	700/19
11	US 20030 05264 5 A1	<input checked="" type="checkbox"/>	Power source circuit, electronic device being equipped with same power source circuit and control method of power source circuit	320/110
12	US 20030 04122 5 A1	<input checked="" type="checkbox"/>	Mechanism for handling load lock/store conditional primitives in directory-based distributed shared memory multiprocessors	712/30
13	US 20020 19906 7 A1	<input checked="" type="checkbox"/>	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
14	US 20020 12076 1 A1	<input checked="" type="checkbox"/>	Method and system for executing protocol stack instructions to form a packet for causing a computing device to perform an operation	709/230
15	US 20020 11660 5 A1	<input checked="" type="checkbox"/>	Method and system for initiating execution of software in response to a state	713/1
16	US 20020 11653 2 A1	<input checked="" type="checkbox"/>	Method and system for communicating an information packet and identifying a data structure	709/246
17	US 20020 11647 5 A1	<input checked="" type="checkbox"/>	Method and system for communicating a request packet in response to a state	709/219

This Page Blank (uspto)

	Docum ent ID	U	Title	Current OR
18	US 20020 11639 7 A1	<input checked="" type="checkbox"/>	Method and system for communicating an information packet through multiple router devices	707/200
19	US 20020 11434 1 A1	<input checked="" type="checkbox"/>	Peer-to-peer enterprise storage	370/428
20	US 20020 11420 0 A1	<input checked="" type="checkbox"/>	System for rapid configuration of a programmable logic device	365/200
21	US 20020 11208 7 A1	<input checked="" type="checkbox"/>	Method and system for establishing a data structure of a connection with a client	709/313
22	US 20020 11208 5 A1	<input checked="" type="checkbox"/>	Method and system for communicating an information packet through multiple networks	709/250
23	US 20020 08784 9 A1	<input checked="" type="checkbox"/>	Full multiprocessor speculation mechanism in a symmetric multiprocessor (smp) System	712/235
24	US 20020 08781 0 A1	<input checked="" type="checkbox"/>	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
25	US 20020 01085 1 A1	<input checked="" type="checkbox"/>	Emulated branch effected by trampoline mechanism	712/244
26	US 20020 01081 4 A1	<input checked="" type="checkbox"/>	Methods and apparatus for providing data transfer control	710/22
27	US 20020 00263 9 A1	<input checked="" type="checkbox"/>	Methods and apparatus for loading a very long instruction word memory	710/22
28	US 20010 04218 7 A1	<input checked="" type="checkbox"/>	VARIABLE ISSUE-WIDTH VLIW PROCESSOR	712/2
29	US 20010 02749 9 A1	<input checked="" type="checkbox"/>	Methods and apparatus for providing direct memory access control	710/26
30	US 20010 02114 2 A1	<input checked="" type="checkbox"/>	Synchronous semiconductor memory device allowing easy and fast test	365/233
31	US 66657 49 B1	<input checked="" type="checkbox"/>	Bus protocol for efficiently transferring vector data	710/29
32	US 66623 64 B1	<input checked="" type="checkbox"/>	System and method for reducing synchronization overhead in multithreaded code	718/102
33	US 66511 51 B2	<input checked="" type="checkbox"/>	MFENCE and LFENCE micro-architectural implementation method and system	711/163
34	US 66437 63 B1	<input checked="" type="checkbox"/>	Register pipe for multi-processing engine environment	712/11
35	US 66427 43 B2	<input checked="" type="checkbox"/>	System for rapid configuration of a programmable logic device	326/37
36	US 66369 50 B1	<input checked="" type="checkbox"/>	Computer architecture for shared memory access	711/147

FIG. 2

	Docum ent ID	U	Title	Current OR
37	US 66257 20 B1	<input checked="" type="checkbox"/>	System for posting vector synchronization instructions to vector instruction queue to separate vector instructions from different application programs	712/4
38	US 66153 38 B1	<input checked="" type="checkbox"/>	Clustered architecture in a VLIW processor	712/24
39	US 66119 00 B2	<input checked="" type="checkbox"/>	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
40	US 65913 70 B1	<input checked="" type="checkbox"/>	Multinode computer system with distributed clock synchronization system	713/502
41	US 65534 86 B1	<input checked="" type="checkbox"/>	Context switching for vector transfer unit	712/222
42	US 65131 07 B1	<input checked="" type="checkbox"/>	Vector transfer system generating address error exception when vector to be transferred does not start and end on same memory page	712/4
43	US 65078 77 B1	<input checked="" type="checkbox"/>	Asynchronous concurrent dual-stream FIFO	710/53
44	US 65052 96 B2	<input checked="" type="checkbox"/>	Emulated branch effected by trampoline mechanism	712/244
45	US 64842 30 B1	<input checked="" type="checkbox"/>	Method and system for speculatively processing a load instruction before completion of a preceding synchronization instruction	711/100
46	US 64808 18 B1	<input checked="" type="checkbox"/>	Debugging techniques in a multithreaded environment	703/26
47	US 64734 02 B1	<input checked="" type="checkbox"/>	Communications link interconnecting service control points of a load sharing group for traffic management control	370/236
48	US 64669 88 B1	<input checked="" type="checkbox"/>	Multiprocessor synchronization and coherency control system	709/248
49	US 64635 11 B2	<input checked="" type="checkbox"/>	System and method for high performance execution of locked memory instructions in a system with distributed memory and a restrictive memory model	711/145
50	US 64570 73 B2	<input checked="" type="checkbox"/>	Methods and apparatus for providing data transfer control	710/22
51	US 64533 67 B2	<input checked="" type="checkbox"/>	Methods and apparatus for providing direct memory access control	710/26
52	US 64153 80 B1	<input checked="" type="checkbox"/>	Speculative execution of a load instruction by associating the load instruction with a previously executed store instruction	712/217
53	US 63967 68 B2	<input checked="" type="checkbox"/>	Synchronous semiconductor memory device allowing easy and fast test	365/233
54	US 63743 70 B1	<input checked="" type="checkbox"/>	Method and system for flexible control of BIST registers based upon on-chip events	714/39
55	US 63706 25 B1	<input checked="" type="checkbox"/>	Method and apparatus for lock synchronization in a microprocessor system	711/152
56	US 63538 29 B1	<input checked="" type="checkbox"/>	Method and system for memory allocation in a multiprocessing environment	707/100
57	US 63203 50 B1	<input checked="" type="checkbox"/>	Modulation control type of AC machine	318/811
58	US 63145 60 B1	<input checked="" type="checkbox"/>	Method and apparatus for a translation system that aggressively optimizes and preserves full synchronous exception state	717/153
59	US 63144 71 B1	<input checked="" type="checkbox"/>	Techniques for an interrupt free operating system	710/5

!

Printed by HPS Server
for

Walk-Up Printing

Printer: cpk2_2c21_gblrptr
Date: 01/05/04
Time: 10:56:18

Document Listing

Document	Selected Pages	Page Range	Copies
US005442760	14	1 - 14	1
Total (1)	14	-	-

	Docum ent ID	U	Title	Current OR
60	US 62860 95 B1	<input checked="" type="checkbox"/>	Computer apparatus having special instructions to force ordered load and store operations	712/216
61	US 62791 00 B1	<input checked="" type="checkbox"/>	Local stall control method and structure in a microprocessor	712/24
62	US 62720 64 B1	<input checked="" type="checkbox"/>	Memory with combined synchronous burst and bus efficient functionality	365/230 .08
63	US 62600 82 B1	<input checked="" type="checkbox"/>	Methods and apparatus for providing data transfer control	710/22
64	US 62596 47 B1	<input checked="" type="checkbox"/>	Synchronous semiconductor memory device allowing easy and fast test	365/230 .01
65	US 62566 83 B1	<input checked="" type="checkbox"/>	Methods and apparatus for providing direct memory access control	710/26
66	US 62471 72 B1	<input checked="" type="checkbox"/>	Method for a translation system that aggressively optimizes and preserves full synchronous exception state	717/141
67	US 61890 88 B1	<input checked="" type="checkbox"/>	Forwarding stored data fetched for out-of-order load/read operation to over-taken operation read-accessing same memory location	712/216
68	US 61784 44 B1	<input checked="" type="checkbox"/>	System and method that prevent messages transferred among networked data processing systems from becoming out of sequence	709/208
69	US 61759 30 B1	<input checked="" type="checkbox"/>	Demand based sync bus operation	714/3
70	US 61675 09 A	<input checked="" type="checkbox"/>	Branch performance in high speed processor	712/237
71	US 61635 00 A	<input checked="" type="checkbox"/>	Memory with combined synchronous burst and bus efficient functionality	365/230 .08
72	US 61118 07 A	<input checked="" type="checkbox"/>	Synchronous semiconductor memory device allowing easy and fast text	365/230 .01
73	US 60790 12 A	<input checked="" type="checkbox"/>	Computer that selectively forces ordered execution of store and load operations between a CPU and a shared memory	712/216
74	US 60761 58 A	<input checked="" type="checkbox"/>	Branch prediction in high-performance processor	712/230
75	US 60650 86 A	<input checked="" type="checkbox"/>	Demand based sync bus operation	710/310
76	US 60584 06 A	<input checked="" type="checkbox"/>	Variable length fractional bandwidth low-pass filtering	708/313
77	US 60095 39 A	<input checked="" type="checkbox"/>	Cross-triggering CPUs for enhanced test operations in a multi-CPU computer system	714/30
78	US 60031 07 A	<input checked="" type="checkbox"/>	Circuitry for providing external access to signals that are internal to an integrated circuit chip package	710/316
79	US 60029 40 A	<input checked="" type="checkbox"/>	Mobile radio station	455/502
80	US 59957 46 A	<input checked="" type="checkbox"/>	Byte-compare operation for high-performance processor	712/220
81	US 59783 52 A	<input checked="" type="checkbox"/>	Multiplex transmission system	370/216
82	US 59783 11 A	<input checked="" type="checkbox"/>	Memory with combined synchronous burst and bus efficient functionality	365/233

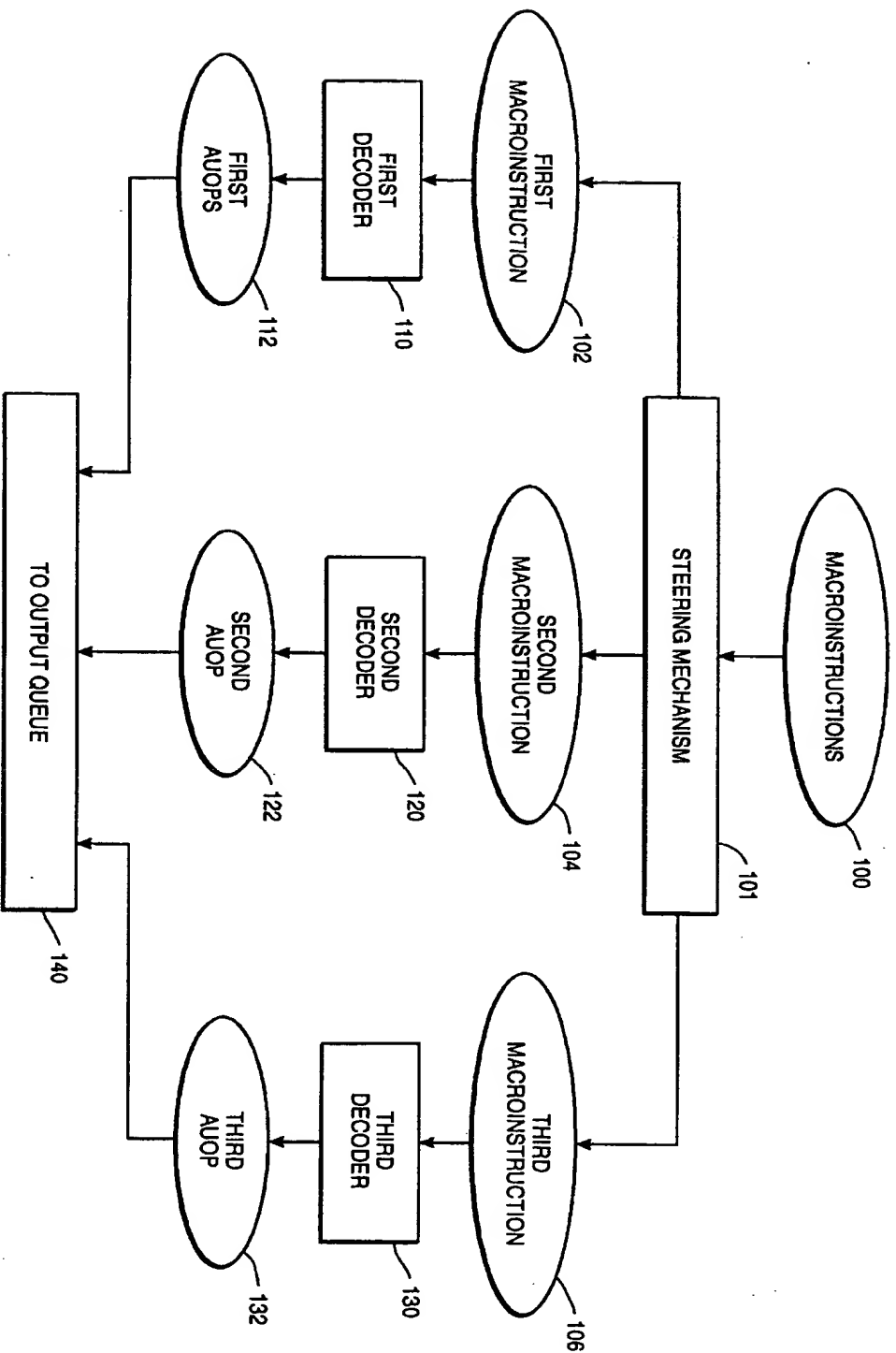


FIG. 1

	Docum ent ID	U	Title	Current OR
83	US 59564 77 A	<input checked="" type="checkbox"/>	Method for processing information in a microprocessor to facilitate debug and performance monitoring	714/30
84	US 59564 76 A	<input checked="" type="checkbox"/>	Circuitry and method for detecting signal patterns on a bus using dynamically changing expected patterns	714/30
85	US 59532 35 A	<input checked="" type="checkbox"/>	Method for processing a hardware independent user description to generate logic circuit elements including flip-flops, latches, and three-state buffers and combinations thereof	716/18
86	US 59502 28 A	<input checked="" type="checkbox"/>	Variable-grained memory sharing for clusters of symmetric multi-processors using private and shared state tables	711/148
87	US 59374 35 A	<input checked="" type="checkbox"/>	System and method for skip-sector mapping in a data recording disk drive	711/202
88	US 59335 98 A	<input checked="" type="checkbox"/>	Method for sharing variable-grained memory of workstations by sending particular block including line and size of the block to exchange shared data structures	709/216
89	US 59296 15 A	<input checked="" type="checkbox"/>	Step-up/step-down voltage regulator using an MOS synchronous rectifier	323/224
90	US 59128 86 A	<input checked="" type="checkbox"/>	Digital mobile communication system capable of establishing mutual synchronization among a plurality of radio base stations	370/350
91	US 59059 67 A	<input checked="" type="checkbox"/>	Timing generator with multiple coherent synchronized clocks	702/118
92	US 58870 03 A	<input checked="" type="checkbox"/>	Apparatus and method for comparing a group of binary fields with an expected pattern to generate match results	714/736
93	US 58840 61 A	<input checked="" type="checkbox"/>	Apparatus to perform source operand dependency analysis perform register renaming and provide rapid pipeline recovery for a microprocessor capable of issuing and executing multiple instructions out-of-order in a single processor cycle	712/217
94	US 58812 62 A	<input checked="" type="checkbox"/>	Method and apparatus for blocking execution of and storing load operations during their execution	712/216
95	US 58812 24 A	<input checked="" type="checkbox"/>	Apparatus and method for tracking events in a microprocessor that can retire more than one instruction during a clock cycle	714/47
96	US 58812 17 A	<input checked="" type="checkbox"/>	Input comparison circuitry and method for a programmable state machine	714/30
97	US 58806 71 A	<input checked="" type="checkbox"/>	Flexible circuitry and method for detecting signal patterns on a bus	340/146 .2
98	US 58676 44 A	<input checked="" type="checkbox"/>	System and method for on-chip debug support and performance monitoring in a microprocessor	714/39
99	US 58359 98 A	<input checked="" type="checkbox"/>	Logic cell for programmable logic devices	326/40
100	US 58261 09 A	<input checked="" type="checkbox"/>	Method and apparatus for performing multiple load operations to the same memory location in a computer system	710/39
101	US 58025 85 A	<input checked="" type="checkbox"/>	Batched checking of shared memory accesses	711/154
102	US 57933 86 A	<input checked="" type="checkbox"/>	Register set reordering for a graphics processor based upon the type of primitive to be rendered	345/553
103	US 57874 80 A	<input checked="" type="checkbox"/>	Lock-up free data sharing	711/148
104	US 57784 23 A	<input checked="" type="checkbox"/>	Prefetch instruction for improving performance in reduced instruction set processor	711/118

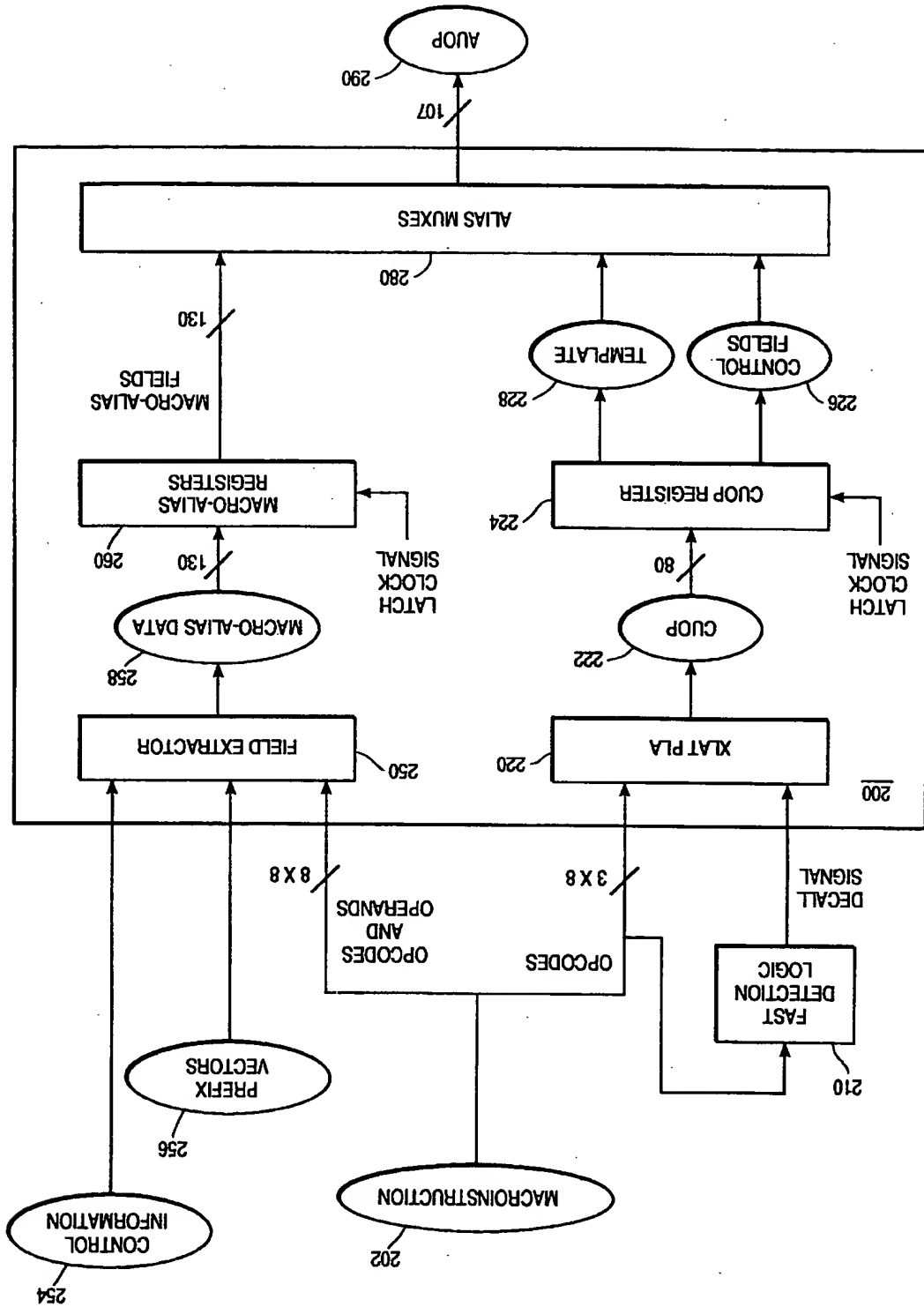


FIG. 2

	Document ID	U	Title	Current OR
105	US 57680 44 A	<input checked="" type="checkbox"/>	Zoned recording embedded servo disk drive having no data identification fields and reduced rotational latency	360/77.08
106	US 57617 29 A	<input checked="" type="checkbox"/>	Validation checking of shared memory accesses	711/148
107	US 57581 83 A	<input checked="" type="checkbox"/>	Method of reducing the number of overhead instructions by modifying the program to locate instructions that access shared data stored at target addresses before program execution	710/5
108	US 57484 88 A	<input checked="" type="checkbox"/>	Method for generating a logic circuit from a hardware independent user description using assignment conditions	716/18
109	US 57427 80 A	<input checked="" type="checkbox"/>	Dual pipeline superscalar reduced instruction set computer system architecture	712/206
110	US 57375 74 A	<input checked="" type="checkbox"/>	Method for generating a logic circuit from a hardware independent user description using mux conditions and hardware selectors	711/162
111	US 57367 95 A	<input checked="" type="checkbox"/>	Solid state AC switch with self-synchronizing means for stealing operating power	307/130
112	US 57245 36 A	<input checked="" type="checkbox"/>	Method and apparatus for blocking execution of and storing load operations during their execution	712/216
113	US 56945 74 A	<input checked="" type="checkbox"/>	Method and apparatus for performing load operations in a computer system	711/140
114	US 56921 53 A	<input checked="" type="checkbox"/>	Method and system for verifying execution order within a multiprocessor data processing system	711/141
115	US 56803 18 A	<input checked="" type="checkbox"/>	Synthesizer for generating a logic network using a hardware independent description	716/18
116	US 56665 06 A	<input checked="" type="checkbox"/>	Apparatus to dynamically control the out-of-order execution of load/store instructions in a processor capable of dispatching, issuing and executing multiple instructions in a single processor cycle	712/216
117	US 56616 61 A	<input checked="" type="checkbox"/>	Method for processing a hardware independent user description to generate logic circuit elements including flip-flops, latches, and three-state buffers and combinations thereof	716/18
118	US 56363 74 A	<input checked="" type="checkbox"/>	Method and apparatus for performing operations based upon the addresses of microinstructions	712/230
119	US 56257 89 A	<input checked="" type="checkbox"/>	Apparatus for source operand dependency analyses register renaming and rapid pipeline recovery in a microprocessor that issues and executes multiple instructions out-of-order in a single cycle	712/217
120	US 56153 50 A	<input checked="" type="checkbox"/>	Apparatus to dynamically control the out-of-order execution of load-store instructions in a processor capable of dispatching, issuing and executing multiple instructions in a single processor cycle	712/218
121	US 56030 47 A	<input checked="" type="checkbox"/>	Superscalar microprocessor architecture	712/23
122	US 56013 95 A	<input checked="" type="checkbox"/>	Organizer system and method for a rotatable storage structure	414/807
123	US 55817 81 A	<input checked="" type="checkbox"/>	Synthesizer for generating a logic network using a hardware independent description	716/18
124	US 55686 24 A	<input checked="" type="checkbox"/>	Byte-compare operation for high-performance processor	712/223
125	US 55487 35 A	<input checked="" type="checkbox"/>	System and method for asynchronously processing store instructions to I/O space	710/7

224 ↖

VALID BIT <u>310</u>	CONTROL FIELD FOR INDIRECT ACCESS OF OTHER REGISTERS <u>226</u>	OPCODE FIELD <u>330</u>	SRC1 <u>340</u>	SRC2 <u>342</u>	DEST <u>344</u>	IMMEDIATE <u>350</u>	. . .
----------------------------	-----------------------------------------------------------------------------------	-------------------------------	--------------------	--------------------	--------------------	-------------------------	-------

FIG. 3

	Docum ent ID	U	Title	Current OR
126	US 55353 61 A	<input checked="" type="checkbox"/>	Cache block replacement scheme based on directory control bit set/reset and hit/miss basis in a multiheading multiprocessor environment	711/145
127	US 55308 41 A	<input checked="" type="checkbox"/>	Method for converting a hardware independent user description of a logic circuit into hardware components	716/3
128	US 55239 03 A	<input checked="" type="checkbox"/>	Sector architecture for fixed block disk drive	360/77. 08
129	US 55106 89 A	<input checked="" type="checkbox"/>	Air gap flux measurement using stator third harmonic voltage	318/809
130	US 54695 51 A	<input checked="" type="checkbox"/>	Method and apparatus for eliminating branches using conditional move instructions	712/239
131	US 54559 44 A	<input checked="" type="checkbox"/>	Method for managing logging and locking of page free space information in a transaction processing system	707/202
132	US 54540 91 A	<input checked="" type="checkbox"/>	Virtual to physical address translation scheme with granularity hint for identifying subsequent pages to be accessed	711/203
133	US 54427 97 A	<input checked="" type="checkbox"/>	Latency tolerant risc-based multiple processor with event driven locality managers resulting from variable tagging	717/149
134	US 54189 73 A	<input checked="" type="checkbox"/>	Digital computer system with cache controller coordinating both vector and scalar operations	712/3
135	US 54106 82 A	<input checked="" type="checkbox"/>	In-register data manipulation for unaligned byte write using data shift in reduced instruction set processor	712/300
136	US 53814 20 A	<input checked="" type="checkbox"/>	Decoupled scan path interface	714/731
137	US 53773 36 A	<input checked="" type="checkbox"/>	Improved method to prefetch load instruction data	712/207
138	US 53677 05 A	<input checked="" type="checkbox"/>	In-register data manipulation using data shift in reduced instruction set processor	712/41
139	US 53652 28 A	<input checked="" type="checkbox"/>	SYNC-NET- a barrier synchronization apparatus for multi-stage networks	340/2.2 1
140	US 53413 18 A	<input checked="" type="checkbox"/>	System for compression and decompression of video data using discrete cosine transform and coding techniques	708/402
141	US 53349 23 A	<input checked="" type="checkbox"/>	Motor torque control method and apparatus	318/805
142	US 52936 31 A	<input checked="" type="checkbox"/>	Analysis and optimization of array variables in compiler for instruction level parallel processor	717/154
143	US 52936 13 A	<input checked="" type="checkbox"/>	Recovery control register	714/15
144	US 52724 29 A	<input checked="" type="checkbox"/>	Air gap flux measurement using stator third harmonic voltage and uses	318/808
145	US 52708 32 A	<input checked="" type="checkbox"/>	System for compression and decompression of video data using discrete cosine transform and coding techniques	382/246
146	US 52690 17 A	<input checked="" type="checkbox"/>	Type 1, 2 and 3 retry and checkpointing	714/15
147	US 52689 39 A	<input checked="" type="checkbox"/>	Control system and method for a nuclear reactor	376/210
148	US 52652 33 A	<input checked="" type="checkbox"/>	Method and apparatus for providing total and partial store ordering for a memory in multi-processor system	711/118

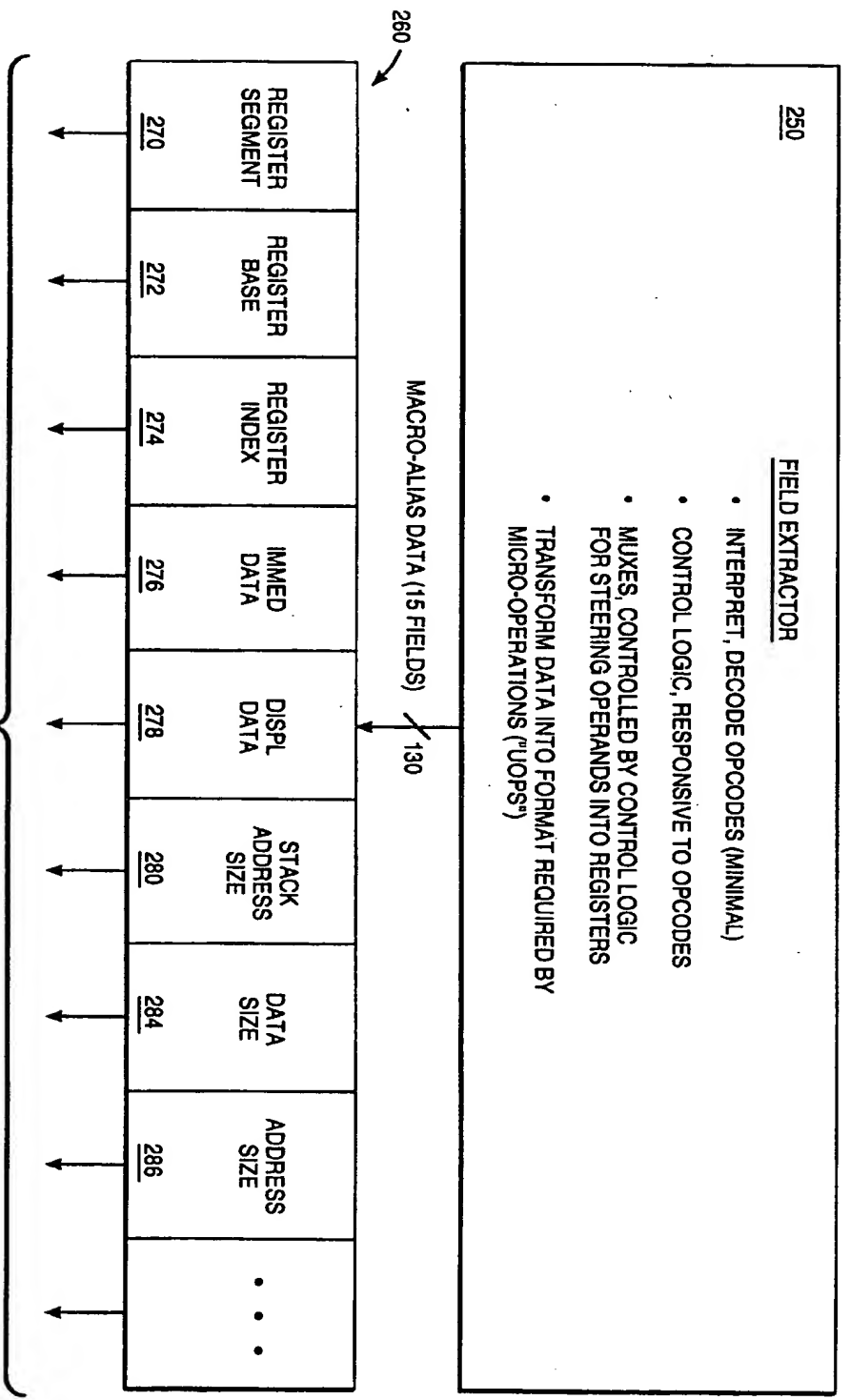
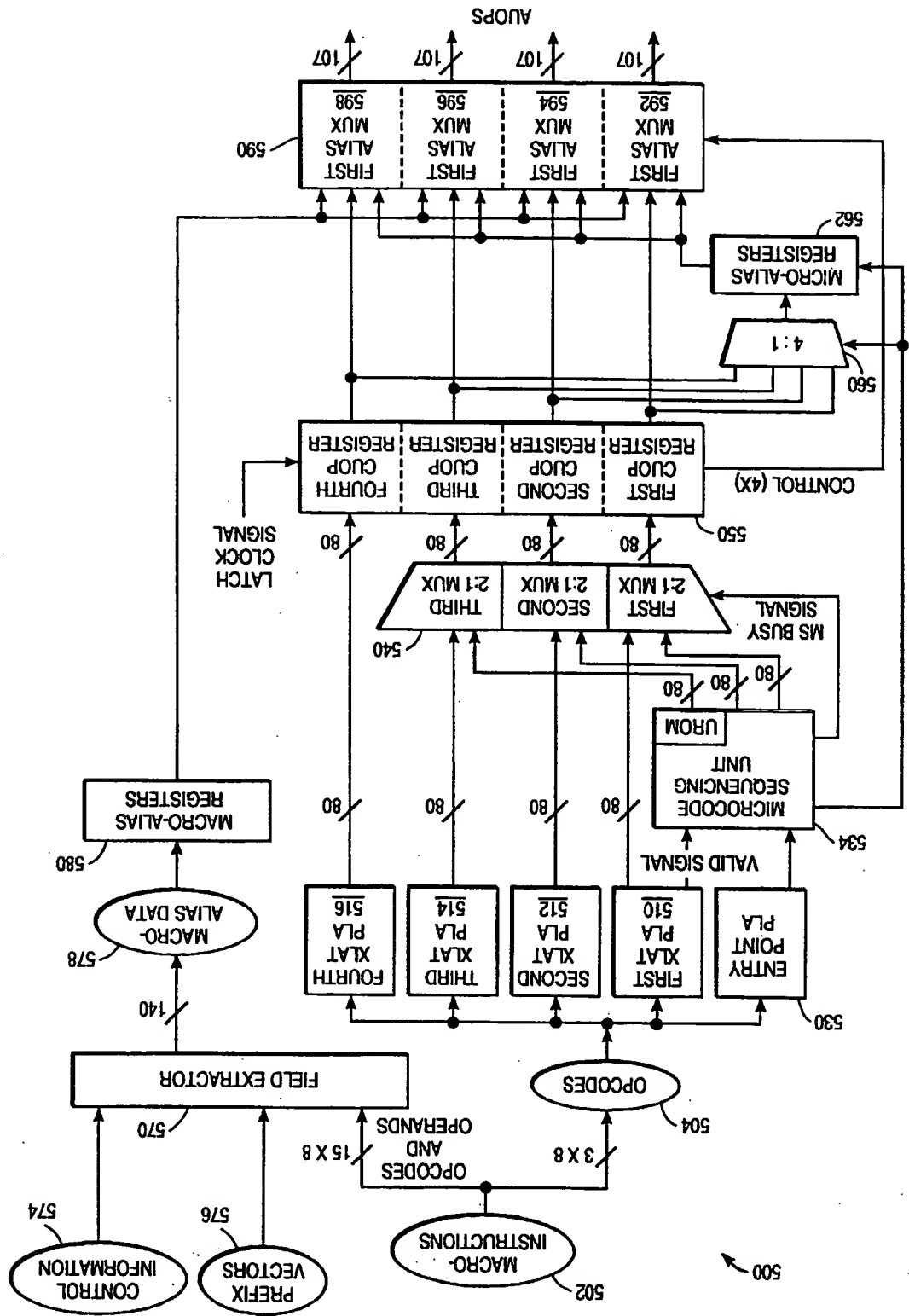
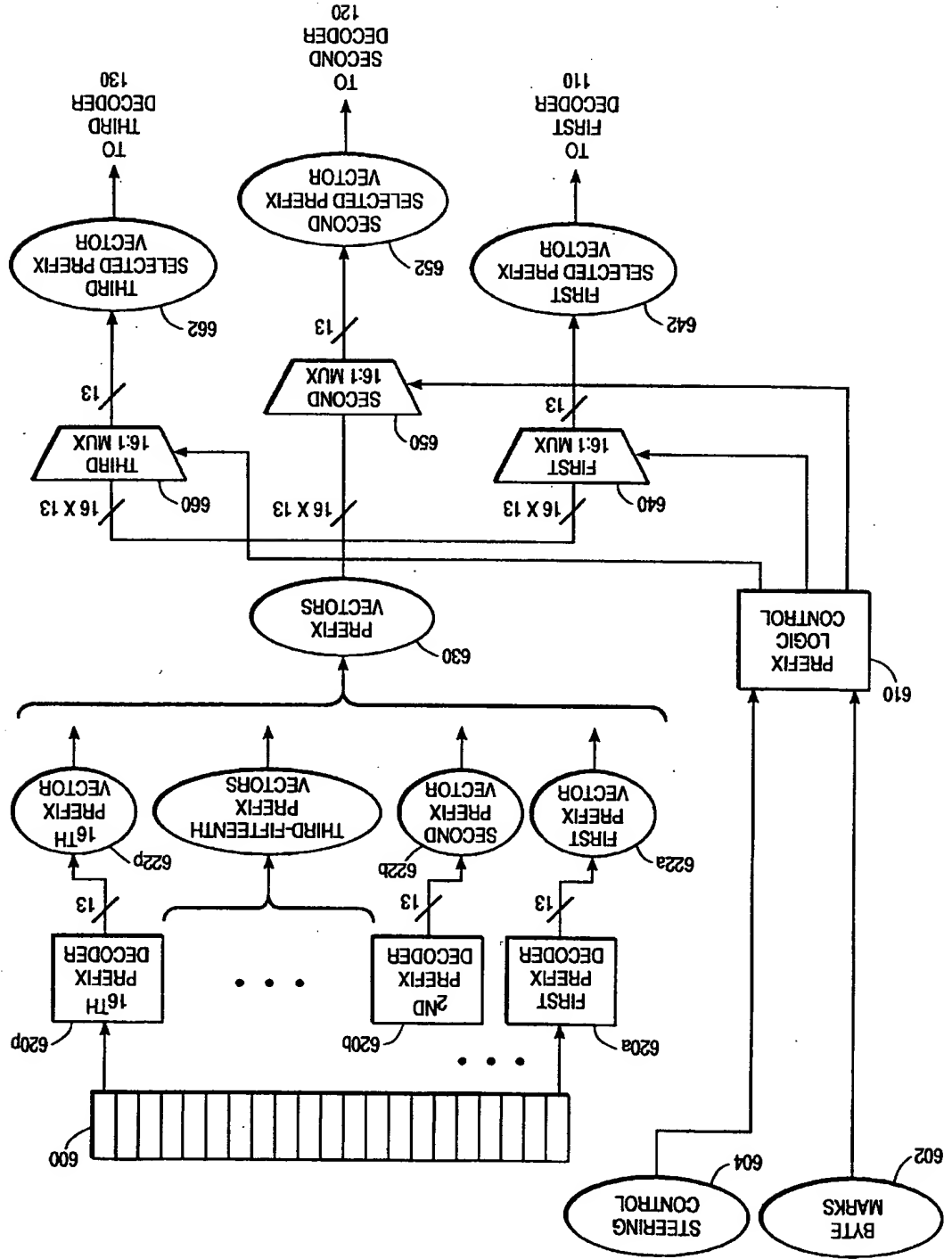


FIG. 4

	Docum ent ID	U	Title	Current OR
149	US 52490 38 A	<input checked="" type="checkbox"/>	System and method for converting component video signals to a composite video signal compatible with the PAL standard	348/453
150	US 51969 46 A	<input checked="" type="checkbox"/>	System for compression and decompression of video data using discrete cosine transform and coding techniques	358/426 .02
151	US 51931 67 A	<input checked="" type="checkbox"/>	Ensuring data integrity by locked-load and conditional-store operations in a multiprocessor system	711/163
152	US 51915 48 A	<input checked="" type="checkbox"/>	System for compression and decompression of video data using discrete cosine transform and coding techniques	708/402
153	US 50559 62 A	<input checked="" type="checkbox"/>	Relay actuation circuitry	361/187
154	US 50217 75 A	<input checked="" type="checkbox"/>	Synchronization method and circuit for display drivers	345/213
155	US 49791 91 A	<input checked="" type="checkbox"/>	Autonomous N-modular redundant fault tolerant clock system	375/357
156	US 49759 60 A	<input checked="" type="checkbox"/>	Electronic facial tracking and detection system and method and apparatus for automated speech recognition	704/251
157	US 49758 33 A	<input checked="" type="checkbox"/>	Multiprocessor system which only allows alternately accessing to shared memory upon receiving read and write request signals	711/152
158	US 48470 39 A	<input checked="" type="checkbox"/>	Steam chest crossties for improved turbine operations	376/297
159	US 48354 80 A	<input checked="" type="checkbox"/>	Electronic signal synchronization apparatus for radar and the like	342/135
160	US 47945 21 A	<input checked="" type="checkbox"/>	Digital computer with cache capable of concurrently handling multiple accesses from parallel processors	711/130
161	US 47837 36 A	<input checked="" type="checkbox"/>	Digital computer with multisection cache	711/130
162	US 46912 31 A	<input checked="" type="checkbox"/>	Bottle inspection system	348/127
163	US 46137 60 A	<input checked="" type="checkbox"/>	Power generating equipment	290/1C
164	US 44777 61 A	<input checked="" type="checkbox"/>	Method of and system for minimizing current consumption of one or more A-C motors driving a variable load	318/800
165	US 43862 82 A	<input checked="" type="checkbox"/>	Emitter function logic (EFL) shift register	377/81
166	US 42700 54 A	<input checked="" type="checkbox"/>	Power plant	290/4R
167	US 42584 24 A	<input checked="" type="checkbox"/>	System and method for operating a steam turbine and an electric power generating plant	700/290
168	US 42271 44 A	<input checked="" type="checkbox"/>	Error compensation of synchro control transmitters	323/348
169	US 41722 81 A	<input checked="" type="checkbox"/>	Microprogrammable control processor for a minicomputer or the like	712/221
170	US 41532 06 A	<input checked="" type="checkbox"/>	Crushing process for recyclable plastic containers	241/14
171	US 40644 85 A	<input checked="" type="checkbox"/>	Digital load control circuit and method for power monitoring and limiting system	307/39



	Docum ent ID	U	Title	Current OR
172	US 40578 36 A	<input checked="" type="checkbox"/>	Slow scan television scan converter	348/22
173	US 40055 81 A	<input checked="" type="checkbox"/>	Method and apparatus for controlling a steam turbine	60/660
174	US 39400 01 A	<input checked="" type="checkbox"/>	Recyclable plastic containers	215/12. 2
175	US 39315 56 A	<input checked="" type="checkbox"/>	System for driving a direct-current motor in synchronism with an external signal	388/812
176	US 39241 40 A	<input checked="" type="checkbox"/>	System for monitoring and controlling industrial gas turbine power plants including facility for dynamic calibration control instrumentation	290/40R
177	US 37725 79 A	<input checked="" type="checkbox"/>	CONTROL MEANS FOR HIGH SPEED HOIST	318/742
178	US 35607 96 A	<input type="checkbox"/>	RELAY CONTROL SYSTEM FOR PREVENTION OF CONTACT EROSION	361/6



	Docum ent ID	U	Title	Current OR
1	US 20030 18246 5 A1	<input type="checkbox"/>	Lock-free implementation of dynamic-sized shared data structure	709/314
2	US 20030 18246 2 A1	<input checked="" type="checkbox"/>	Value recycling facility for multithreaded computations	709/310
3	US 20030 17457 2 A1	<input checked="" type="checkbox"/>	Non-blocking memory management mechanism for supporting dynamic-sized data structures	365/230 .03
4	US 20030 15402 8 A1	<input checked="" type="checkbox"/>	Tracing multiple data access instructions	702/1
5	US 20030 14029 1 A1	<input checked="" type="checkbox"/>	Method and apparatus for providing JTAG functionality in a remote server management controller	714/724
6	US 20030 14008 5 A1	<input checked="" type="checkbox"/>	Single-word lock-free reference counting	709/107
7	US 20030 07911 3 A1	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/205
8	US 20030 07006 0 A1	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/23
9	US 20030 05608 7 A1	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/207
10	US 20030 05608 6 A1	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/207
11	US 20030 04800 6 A1	<input checked="" type="checkbox"/>	Uninterruptible power supply	307/64
12	US 20030 03351 0 A1	<input checked="" type="checkbox"/>	Methods and apparatus for controlling speculative execution of instructions based on a multiaccess memory condition	712/235
13	US 20020 15696 2 A1	<input checked="" type="checkbox"/>	Microprocessor having improved memory management unit and cache memory	711/3
14	US 20020 12933 9 A1	<input checked="" type="checkbox"/>	Parallelism performance analysis based on execution trace information	717/127
15	US 20020 11214 6 A1	<input checked="" type="checkbox"/>	Method and apparatus for synchronizing load operation	712/219
16	US 20020 02932 8 A1	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/23
17	US 20020 00264 0 A1	<input checked="" type="checkbox"/>	Methods and apparatus for providing bit-reversal and multicast functions utilizing DMA controller	710/22

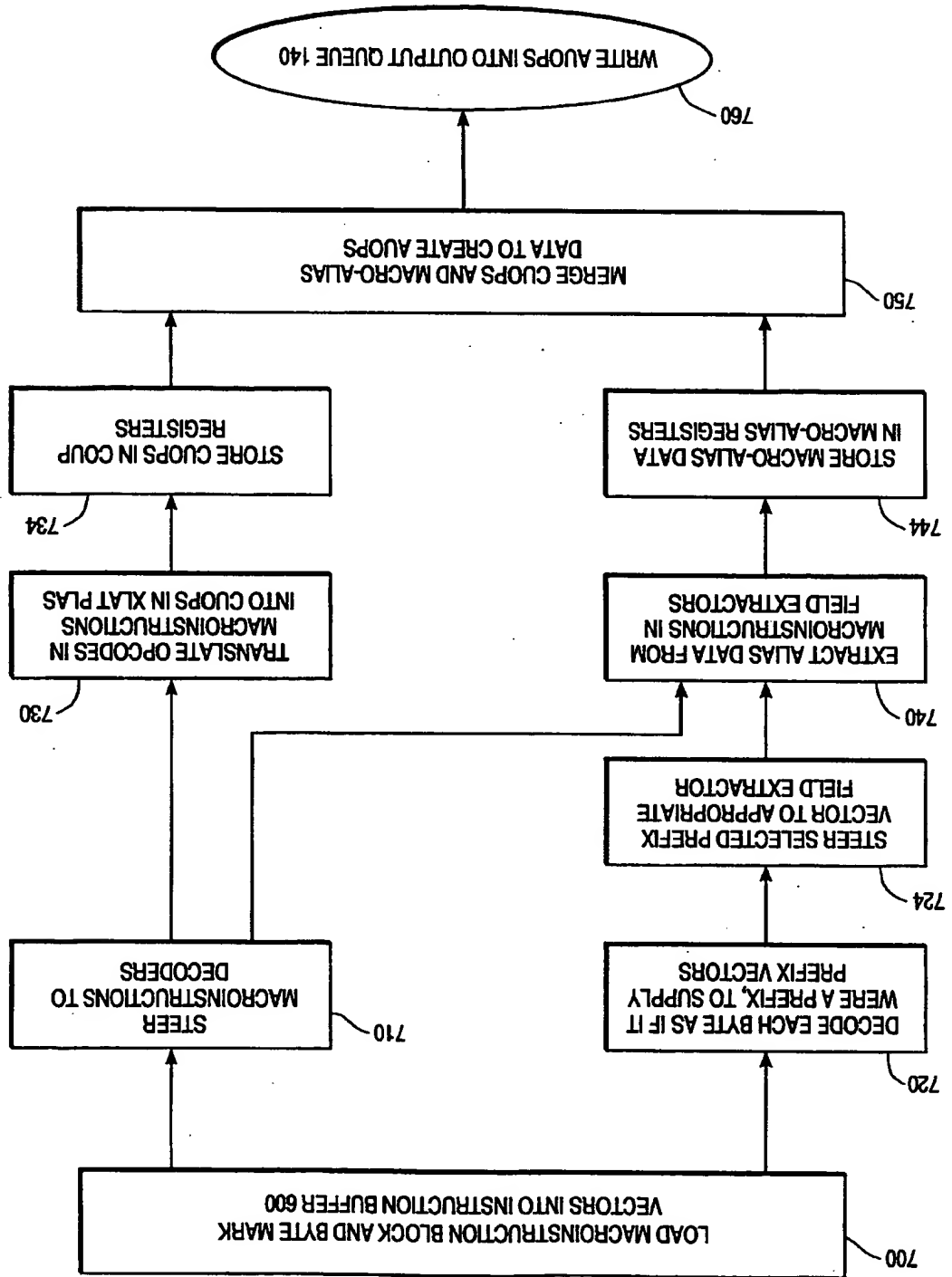


FIG. 7

	Docum ent ID	U	Title	Current OR
18	US 20010 04745 7 A1	<input checked="" type="checkbox"/>	Digital data processor with improved paging	711/141
19	US 20010 03743 4 A1	<input checked="" type="checkbox"/>	Store to load forwarding using a dependency link file	711/146
20	US 20010 03481 9 A1	<input checked="" type="checkbox"/>	Interleaved data path and output management architecture for an interleaved memory and load pulser circuit for outputting the read data	711/157
21	US 20010 03324 5 A1	<input checked="" type="checkbox"/>	Interleaved memory device for burst type access in synchronous read mode with the two semi-arrays independently readable in random access asynchronous mode	341/200
22	US 20010 02709 6 A1	<input checked="" type="checkbox"/>	Method for interrupting an idle state of a communication unit in a communication system, especially in a radio communication system	455/343 .1
23	US 66474 85 B2	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/23
24	US 66292 07 B1	<input checked="" type="checkbox"/>	Method for loading instructions or data into a locked way of a cache memory	711/125
25	US 66256 60 B1	<input checked="" type="checkbox"/>	Multiprocessor speculation mechanism for efficiently managing multiple barrier operations	709/248
26	US 66222 31 B2	<input checked="" type="checkbox"/>	Method and apparatus for paging data and attributes including an atomic attribute	711/209
27	US 66091 92 B1	<input checked="" type="checkbox"/>	System and method for asynchronously overlapping storage barrier operations with old and new storage operations	712/216
28	US 66067 02 B1	<input checked="" type="checkbox"/>	Multiprocessor speculation mechanism with imprecise recycling of storage operations	712/218
29	US 66009 59 B1	<input checked="" type="checkbox"/>	Method and apparatus for implementing microprocessor control logic using dynamic programmable logic arrays	700/7
30	US 65981 28 B1	<input checked="" type="checkbox"/>	Microprocessor having improved memory management unit and cache memory	711/144
31	US 65913 40 B2	<input checked="" type="checkbox"/>	Microprocessor having improved memory management unit and cache memory	711/118
32	US 65913 21 B1	<input checked="" type="checkbox"/>	Multiprocessor system bus protocol with group addresses, responses, and priorities	710/110
33	US 65879 13 B2	<input checked="" type="checkbox"/>	Interleaved memory device for burst type access in synchronous read mode with the two semi-arrays independently readable in random access asynchronous mode	711/5
34	US 65534 60 B1	<input checked="" type="checkbox"/>	Microprocessor having improved memory management unit and cache memory	711/125
35	US 65499 90 B2	<input checked="" type="checkbox"/>	Store to load forwarding using a dependency link file	711/146
36	US 65464 62 B1	<input checked="" type="checkbox"/>	CLFLUSH micro-architectural implementation method and system	711/135
37	US 65264 81 B1	<input checked="" type="checkbox"/>	Adaptive cache coherence protocols	711/147
38	US 65052 77 B1	<input checked="" type="checkbox"/>	Method for just-in-time delivery of load data by intervening caches	711/158

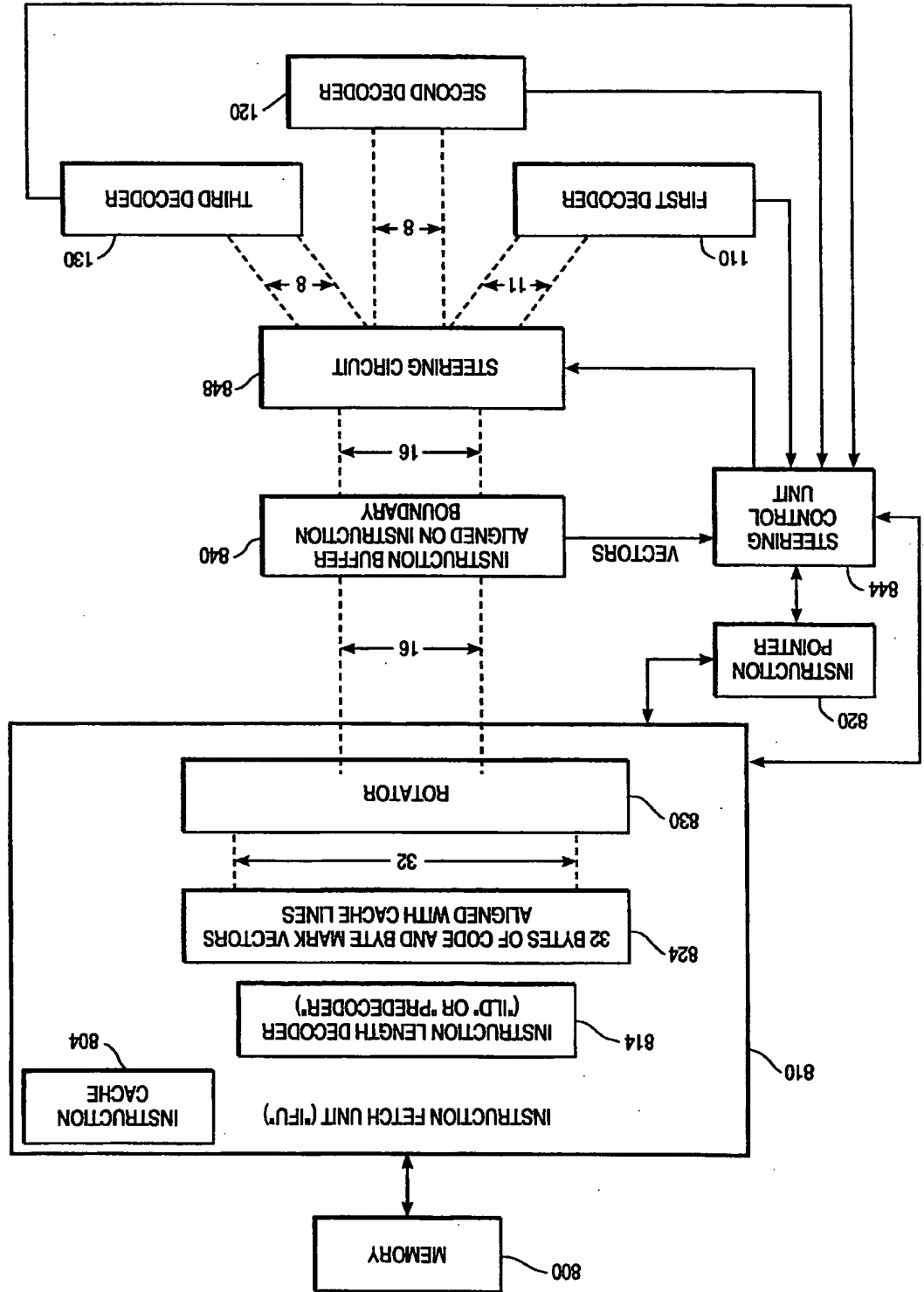
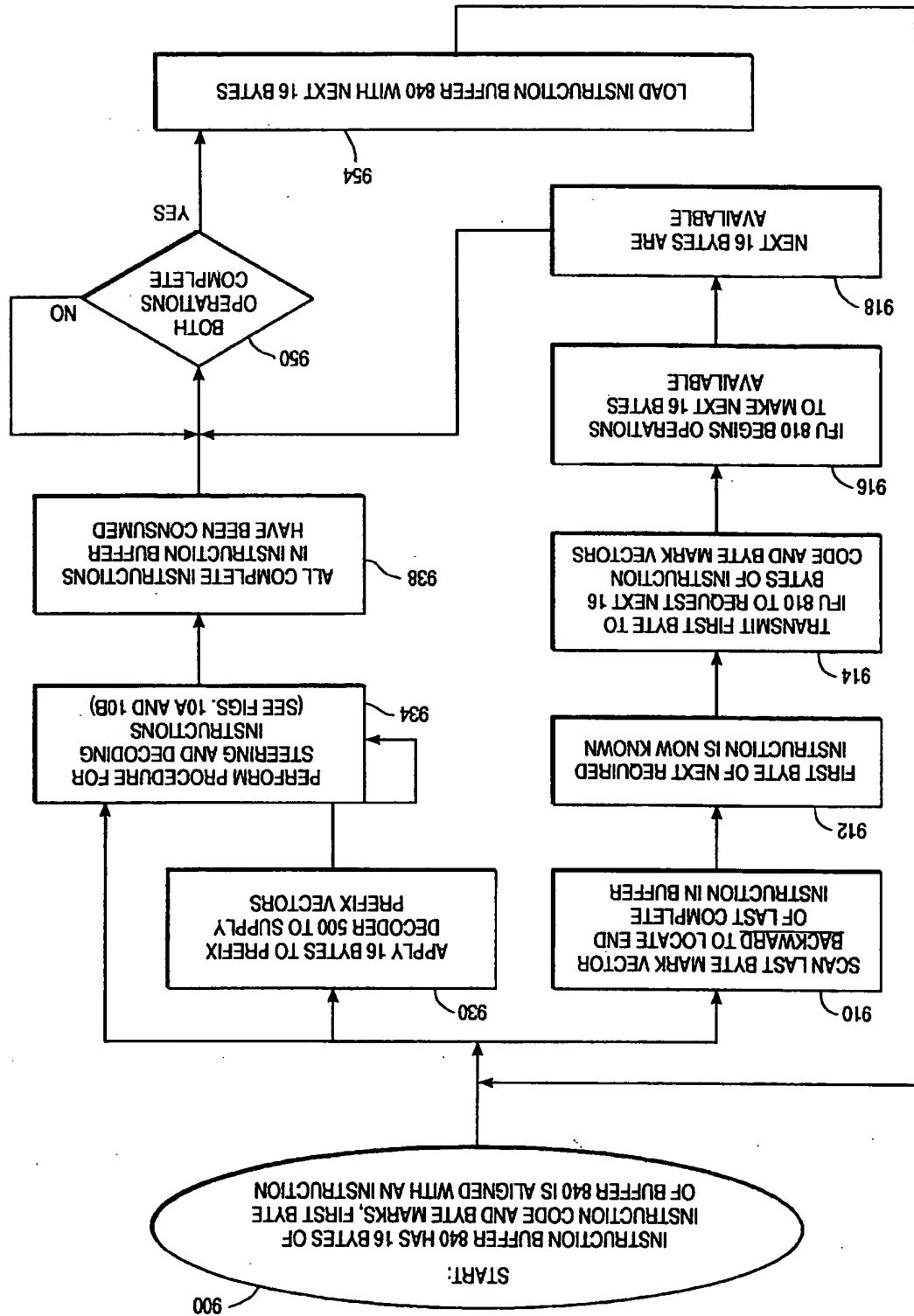


FIG. 8

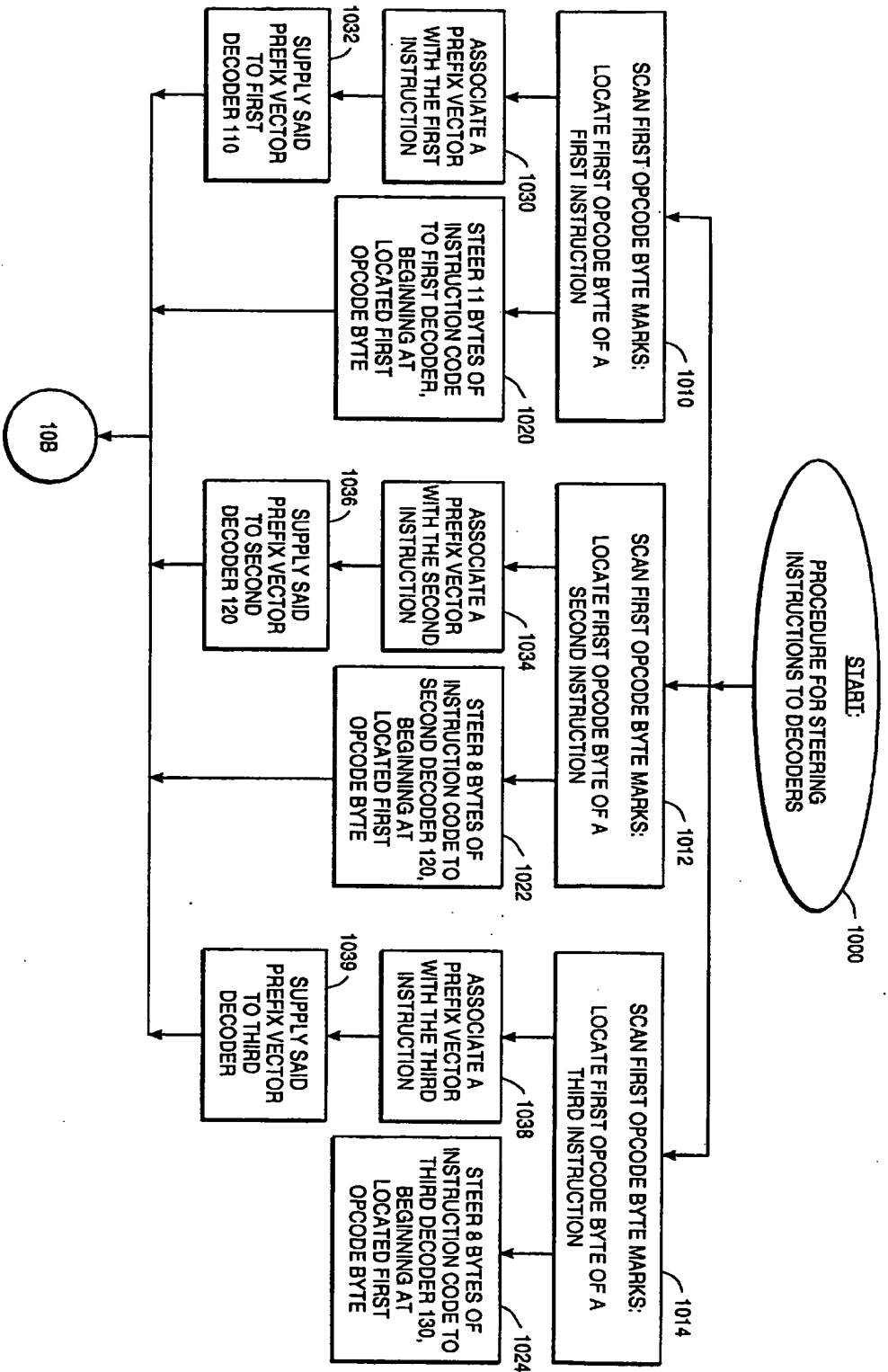
	Document ID	U	Title	Current OR
39	US 64969 40 B1	<input checked="" type="checkbox"/>	Multiple processor system with standby sparing	714/4
40	US 64738 37 B1	<input checked="" type="checkbox"/>	Snoop resynchronization mechanism to preserve read ordering	711/146
41	US 64738 32 B1	<input checked="" type="checkbox"/>	Load/store unit having pre-cache and post-cache queues for low latency load memory operations	711/118
42	US 64704 31 B2	<input checked="" type="checkbox"/>	Interleaved data path and output management architecture for an interleaved memory and load pulser circuit for outputting the read data	711/157
43	US 64601 24 B1	<input checked="" type="checkbox"/>	Method of using delays to speed processing of inferred critical program portions	711/163
44	US 64300 74 B1	<input checked="" type="checkbox"/>	Selective look-ahead match line pre-charging in a partitioned content addressable memory array	365/49
45	US 64271 93 B1	<input checked="" type="checkbox"/>	Deadlock avoidance using exponential backoff	711/146
46	US 64153 60 B1	<input checked="" type="checkbox"/>	Minimizing self-modifying code checks for uncacheable memory types	711/139
47	US 64120 43 B1	<input checked="" type="checkbox"/>	Microprocessor having improved memory management unit and cache memory	711/118
48	US 63490 49 B1	<input checked="" type="checkbox"/>	High speed low power content addressable memory	365/49
49	US 63321 85 B1	<input checked="" type="checkbox"/>	Method and apparatus for paging data and attributes including an atomic attribute for digital data processor	711/209
50	US 63112 61 B1	<input checked="" type="checkbox"/>	Apparatus and method for improving superscalar processors	712/23
51	US 62726 19 B1	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/41
52	US 62667 44 B1	<input checked="" type="checkbox"/>	Store to load forwarding using a dependency link file	711/146
53	US 62567 20 B1	<input checked="" type="checkbox"/>	High performance, superscalar-based computer system with out-of-order instruction execution	712/23
54	US 62432 80 B1	<input checked="" type="checkbox"/>	Selective match line pre-charging in a partitioned content addressable memory array	365/49
55	US 62337 02 B1	<input checked="" type="checkbox"/>	Self-checked, lock step processor pairs	714/48
56	US 61579 67 A	<input checked="" type="checkbox"/>	Method of data communication flow control in a data processing system using busy/ready commands	710/19
57	US 61516 89 A	<input checked="" type="checkbox"/>	Detecting and isolating errors occurring in data communication in a multiple processor system	714/49
58	US 61417 32 A	<input checked="" type="checkbox"/>	Burst-loading of instructions into processor cache by execution of linked jump instructions embedded in cache line size blocks	711/137
59	US 61287 23 A	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/23
60	US 61120 19 A	<input checked="" type="checkbox"/>	Distributed instruction queue	712/214
61	US 61015 94 A	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/41

FIG. 9



	Docum ent ID	U	Title	Current OR
62	US 60921 81 A	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/206
63	US 60887 58 A	<input checked="" type="checkbox"/>	Method and apparatus for distributing data in a digital data processor with distributed memory	711/100
64	US 60848 09 A	<input checked="" type="checkbox"/>	Main amplifier circuit and input-output bus for a dynamic random access memory	365/203
65	US 60732 10 A	<input checked="" type="checkbox"/>	Synchronization of weakly ordered write combining operations using a fencing mechanism	711/118
66	US 60386 54 A	<input checked="" type="checkbox"/>	High performance, superscalar-based computer system with out-of-order instruction execution	712/23
67	US 60162 70 A	<input checked="" type="checkbox"/>	Flash memory architecture that utilizes a time-shared address bus scheme and separate memory cell access paths for simultaneous read/write operations	365/185 .11
68	US 60063 18 A	<input checked="" type="checkbox"/>	General purpose, dynamic partitioning, programmable media processor	712/28
69	US 59681 35 A	<input checked="" type="checkbox"/>	Processing instructions up to load instruction after executing sync flag monitor instruction during plural processor shared memory store/load access synchronization	709/400
70	US 59648 35 A	<input checked="" type="checkbox"/>	Storage access validation to data messages using partial storage address data indexed entries containing permissible address range validation for message source	709/216
71	US 59616 29 A	<input checked="" type="checkbox"/>	High performance, superscalar-based computer system with out-of-order instruction execution	712/23
72	US 59532 86 A	<input checked="" type="checkbox"/>	Synchronous DRAM having a high data transfer rate	365/233
73	US 59336 24 A	<input checked="" type="checkbox"/>	Synchronized MIMD multi-processing system and method inhibiting instruction fetch at other processors while one processor services an interrupt	709/400
74	US 59207 16 A	<input checked="" type="checkbox"/>	Compiling a predicated code with direct analysis of the predicated code	717/141
75	US 59149 53 A	<input checked="" type="checkbox"/>	Network message routing using routing table information and supplemental enable information for deadlock prevention	370/392
76	US 58840 60 A	<input checked="" type="checkbox"/>	Processor which performs dynamic instruction scheduling at time of execution within a single clock cycle	712/215
77	US 58812 72 A	<input checked="" type="checkbox"/>	Synchronized MIMD multi-processing system and method inhibiting instruction fetch at other processors on write to program counter of one processor	709/400
78	US 58729 90 A	<input checked="" type="checkbox"/>	Reordering of memory reference operations and conflict resolution via rollback in a multiprocessing environment	712/24
79	US 58675 01 A	<input checked="" type="checkbox"/>	Encoding for communicating data and commands	370/474
80	US 58388 94 A	<input checked="" type="checkbox"/>	Logical, fail-functional, dual central processor units formed from three processor units	714/11
81	US 58226 03 A	<input checked="" type="checkbox"/>	High bandwidth media processor interface for transmitting data in the form of packets with requests linked to associated responses by identification data	712/1
82	US 58093 21 A	<input checked="" type="checkbox"/>	General purpose, multiple precision parallel operation, programmable media processor	712/1
83	US 58092 88 A	<input checked="" type="checkbox"/>	Synchronized MIMD multi-processing system and method inhibiting instruction fetch on memory access stall	709/400
84	US 57940 61 A	<input checked="" type="checkbox"/>	General purpose, multiple precision parallel operation, programmable media processor	712/1

FIG. 10A



	Docum ent ID	U	Title	Current OR
85	US 57940 60 A	<input checked="" type="checkbox"/>	General purpose, multiple precision parallel operation, programmable media processor	712/1
86	US 57907 76 A	<input checked="" type="checkbox"/>	Apparatus for detecting divergence between a pair of duplexed, synchronized processor elements	714/10
87	US 57782 45 A	<input checked="" type="checkbox"/>	Method and apparatus for dynamic allocation of multiple buffers in a processor	712/23
88	US 57655 25 A	<input checked="" type="checkbox"/>	Intake system for an internal combustion engine	123/308
89	US 57519 55 A	<input checked="" type="checkbox"/>	Method of synchronizing a pair of central processor units for duplex, lock-step operation by copying data into a corresponding locations of another memory	714/12
90	US 57519 32 A	<input checked="" type="checkbox"/>	Fail-fast, fail-functional, fault-tolerant multiprocessor system	714/12
91	US 57428 40 A	<input checked="" type="checkbox"/>	General purpose, multiple precision parallel operation, programmable media processor	712/210
92	US 57348 21 A	<input checked="" type="checkbox"/>	Method and apparatus for a direct data transmission between a communication network interface and a multimedia data processor	709/200
93	US 56995 38 A	<input checked="" type="checkbox"/>	Efficient firm consistency support mechanisms in an out-of-order execution superscaler multiprocessor	712/23
94	US 56945 77 A	<input checked="" type="checkbox"/>	Memory conflict buffer for achieving memory disambiguation in compile-time code schedule	711/167
95	US 56919 11 A	<input checked="" type="checkbox"/>	Method for pre-processing a hardware independent description of a logic circuit	716/18
96	US 56897 20 A	<input checked="" type="checkbox"/>	High-performance superscalar-based computer system with out-of-order instruction execution	712/23
97	US 56896 89 A	<input checked="" type="checkbox"/>	Clock circuits for synchronized processor systems having clock generator circuit with a voltage control oscillator producing a clock signal synchronous with a master clock signal	709/400
98	US 56785 21 A	<input checked="" type="checkbox"/>	System and methods for electronic control of an accumulator fuel system	123/447
99	US 56758 07 A	<input checked="" type="checkbox"/>	Interrupt message delivery identified by storage location of received interrupt data	710/260
100	US 56755 79 A	<input checked="" type="checkbox"/>	Method for verifying responses to messages using a barrier message	370/248
101	US 56491 35 A	<input checked="" type="checkbox"/>	Parallel processing system and method using surrogate instructions	712/200
102	US 56405 88 A	<input checked="" type="checkbox"/>	CPU architecture performing dynamic instruction scheduling at time of execution within single clock cycle	712/23
103	US 56299 50 A	<input checked="" type="checkbox"/>	Fault management scheme for a cache memory	714/805
104	US 56278 42 A	<input checked="" type="checkbox"/>	Architecture for system-wide standardized intra-module and inter-module fault testing	714/727
105	US 56131 36 A	<input checked="" type="checkbox"/>	Locality manager having memory and independent code, bus interface logic, and synchronization components for a processing element for intercommunication in a latency tolerant multiple processor	712/28
106	US 55748 49 A	<input checked="" type="checkbox"/>	Synchronized data transmission between elements of a processing system	714/12

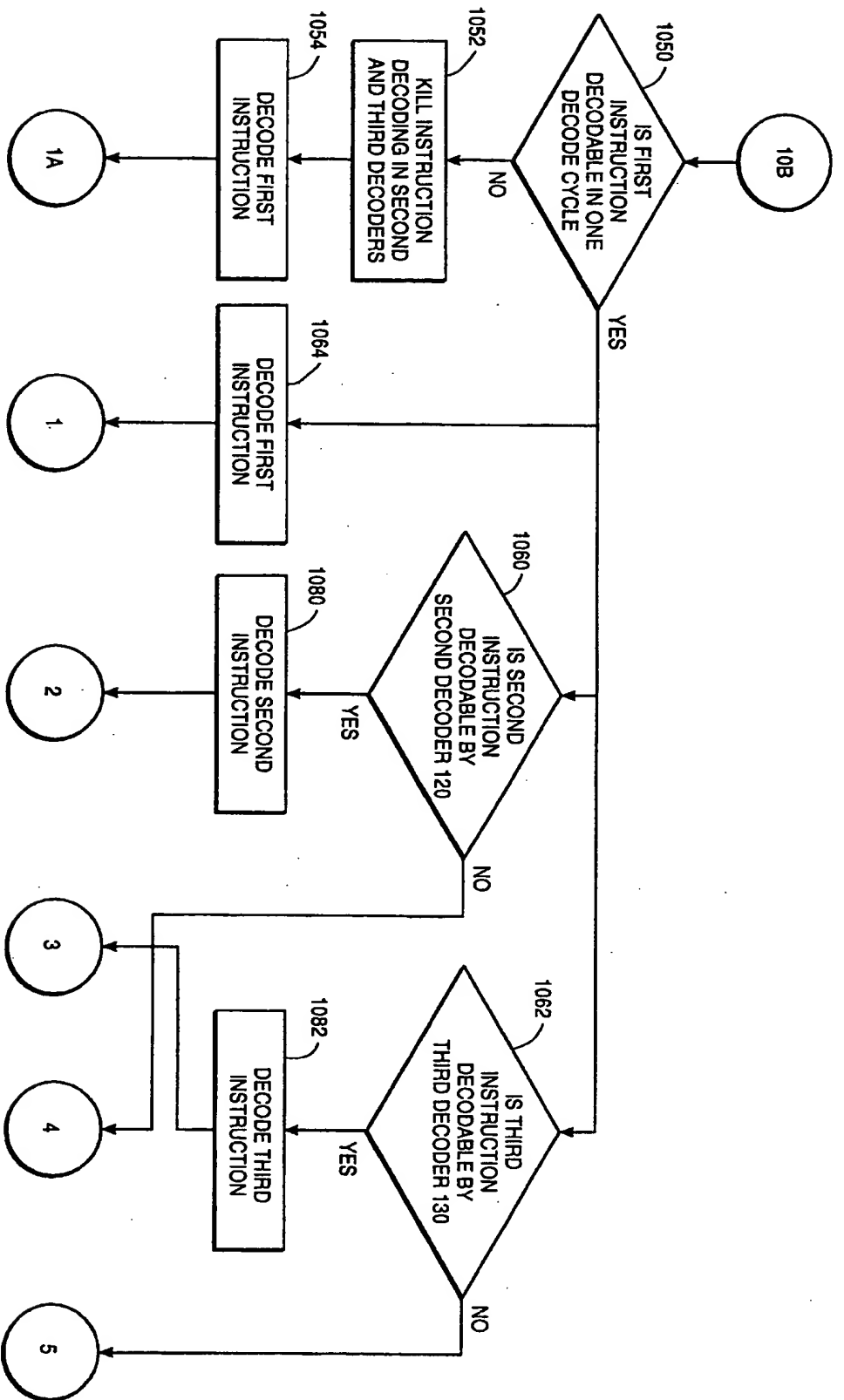


FIG. 10B

	Docum ent ID	U	Title	Current OR
107	US 55600 29 A	<input checked="" type="checkbox"/>	Data processing system with synchronization coprocessor for multiple threads	712/25
108	US 55532 58 A	<input checked="" type="checkbox"/>	Method and apparatus for forming an exchange address for a system with different size caches	711/3
109	US 55530 95 A	<input checked="" type="checkbox"/>	Method and apparatus for exchanging different classes of data during different time intervals	375/222
110	US 55399 11 A	<input checked="" type="checkbox"/>	High-performance, superscalar-based computer system with out-of-order instruction execution	712/23
111	US 55375 49 A	<input checked="" type="checkbox"/>	Communication network with time coordinated station activity by time slot and periodic interval number	709/224
112	US 54935 71 A	<input checked="" type="checkbox"/>	Apparatus and method for digital communications with improved delimiter detection	370/514
113	US 54915 31 A	<input checked="" type="checkbox"/>	Media access controller with a shared class message delivery capability	375/354
114	US 54887 29 A	<input checked="" type="checkbox"/>	Central processing unit architecture with symmetric instruction scheduling to achieve multiple instruction launch and execution	712/209
115	US 54860 80 A	<input checked="" type="checkbox"/>	High speed movement of workpieces in vacuum processing	414/217
116	US 54771 03 A	<input checked="" type="checkbox"/>	Sequence, timing and synchronization technique for servo system controller of a computer disk mass storage device	318/601
117	US 54308 50 A	<input checked="" type="checkbox"/>	Data processing system with synchronization coprocessor for multiple threads	709/314
118	US 54106 21 A	<input checked="" type="checkbox"/>	Image processing system having a sampled filter	382/260
119	US 54003 31 A	<input checked="" type="checkbox"/>	Communication network interface with screeners for incoming messages	370/401
120	US 53414 83 A	<input checked="" type="checkbox"/>	Dynamic hierarchial associative memory	711/206
121	US 53136 47 A	<input checked="" type="checkbox"/>	Digital data processor with improved checkpointing and forking	709/102
122	US 52947 91 A	<input checked="" type="checkbox"/>	System and a method for controlling position of a magnetic head relative to a servo track of a tape by optical detection of an edge of the tape	250/548
123	US 52822 01 A	<input checked="" type="checkbox"/>	Dynamic packet routing network	370/403
124	US 52513 08 A	<input checked="" type="checkbox"/>	Shared memory multiprocessor with data hiding and post-store	711/163
125	US 52260 39 A	<input checked="" type="checkbox"/>	Packet routing switch	370/405
126	US 51465 85 A	<input checked="" type="checkbox"/>	Synchronized fault tolerant clocks for multiprocessor systems	713/400
127	US 50539 83 A	<input checked="" type="checkbox"/>	Filter system having an adaptive control for updating filter samples	708/306
128	US 49775 29 A	<input checked="" type="checkbox"/>	Training simulator for a nuclear power plant	703/18
129	US 49440 36 A	<input checked="" type="checkbox"/>	Signature filter system	367/43

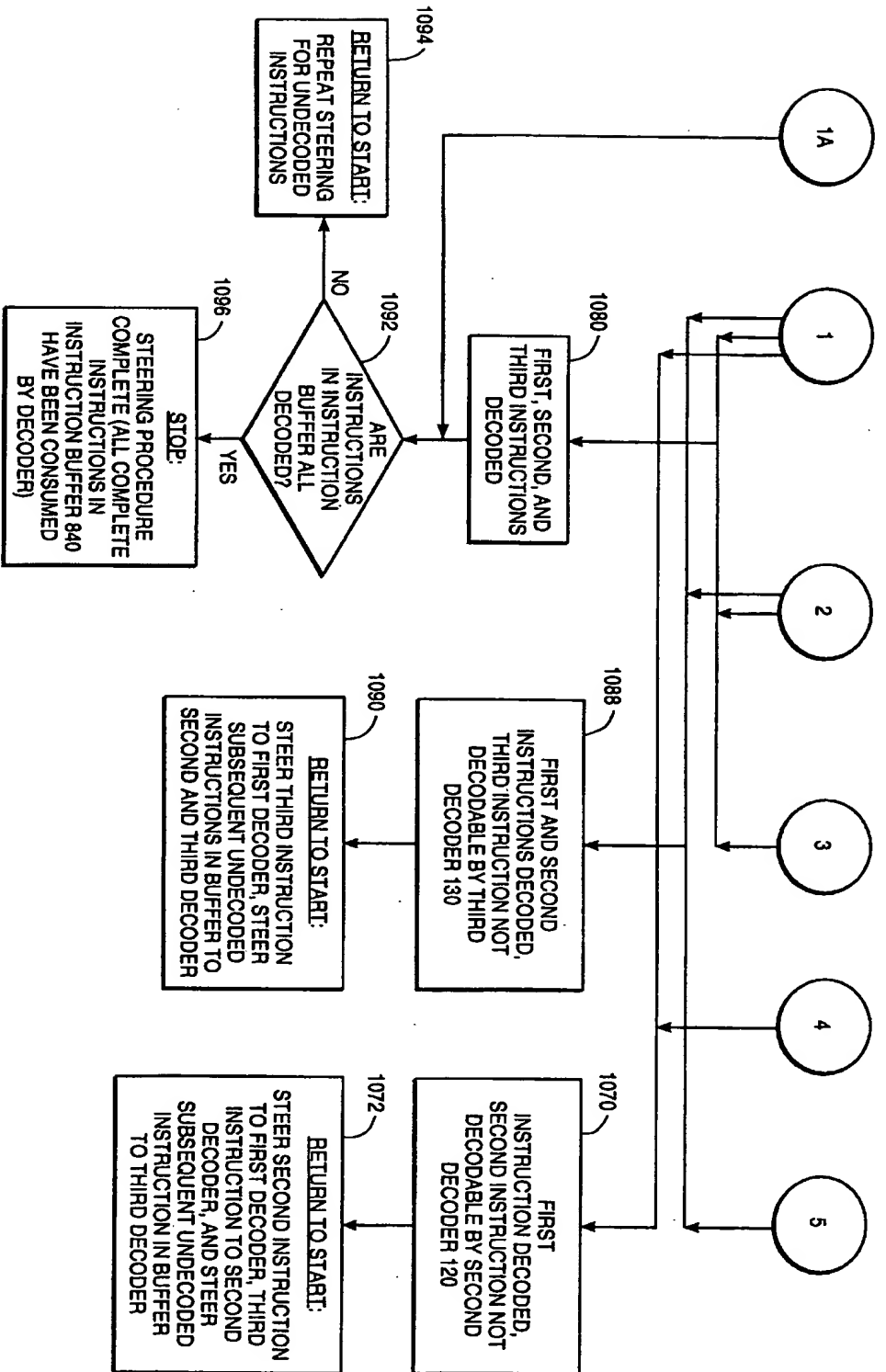


FIG. 100C

	Docum ent ID	U	Title	Current OR
130	US 49222 33 A	<input checked="" type="checkbox"/>	Flow sensor and system incorporating the same for monitoring steam turbine drain valves	340/606
131	US 48751 59 A	<input checked="" type="checkbox"/>	Version management system using plural control fields for synchronizing two versions of files in a multiprocessor system	707/203
132	US 48507 93 A	<input checked="" type="checkbox"/>	Steam chest modifications for improved turbine operations	415/38
133	US 47930 57 A	<input checked="" type="checkbox"/>	Apparatus for mounting power transmission belts on and removing same from pulleys	29/822
134	US 46866 55 A	<input checked="" type="checkbox"/>	Filtering system for processing signature signals	367/59
135	US 45817 15 A	<input checked="" type="checkbox"/>	Fourier transform processor	708/403
136	US 45532 21 A	<input checked="" type="checkbox"/>	Digital filtering system	708/308
137	US 45532 13 A	<input checked="" type="checkbox"/>	Communication system	332/185
138	US 45518 16 A	<input checked="" type="checkbox"/>	Filter display system	708/422
139	US 44919 30 A	<input checked="" type="checkbox"/>	Memory system using filterable signals	708/3
140	US 44357 53 A	<input checked="" type="checkbox"/>	Register allocation system using recursive queuing during source code compilation	717/153
141	US 44266 11 A	<input checked="" type="checkbox"/>	Twelve pulse load commutated inverter drive system	318/803
142	US 42098 43 A	<input checked="" type="checkbox"/>	Method and apparatus for signal enhancement with improved digital filtering	708/422
143	US 39341 28 A	<input checked="" type="checkbox"/>	System and method for operating a steam turbine with improved organization of logic and other functions in a sampled data control	700/290

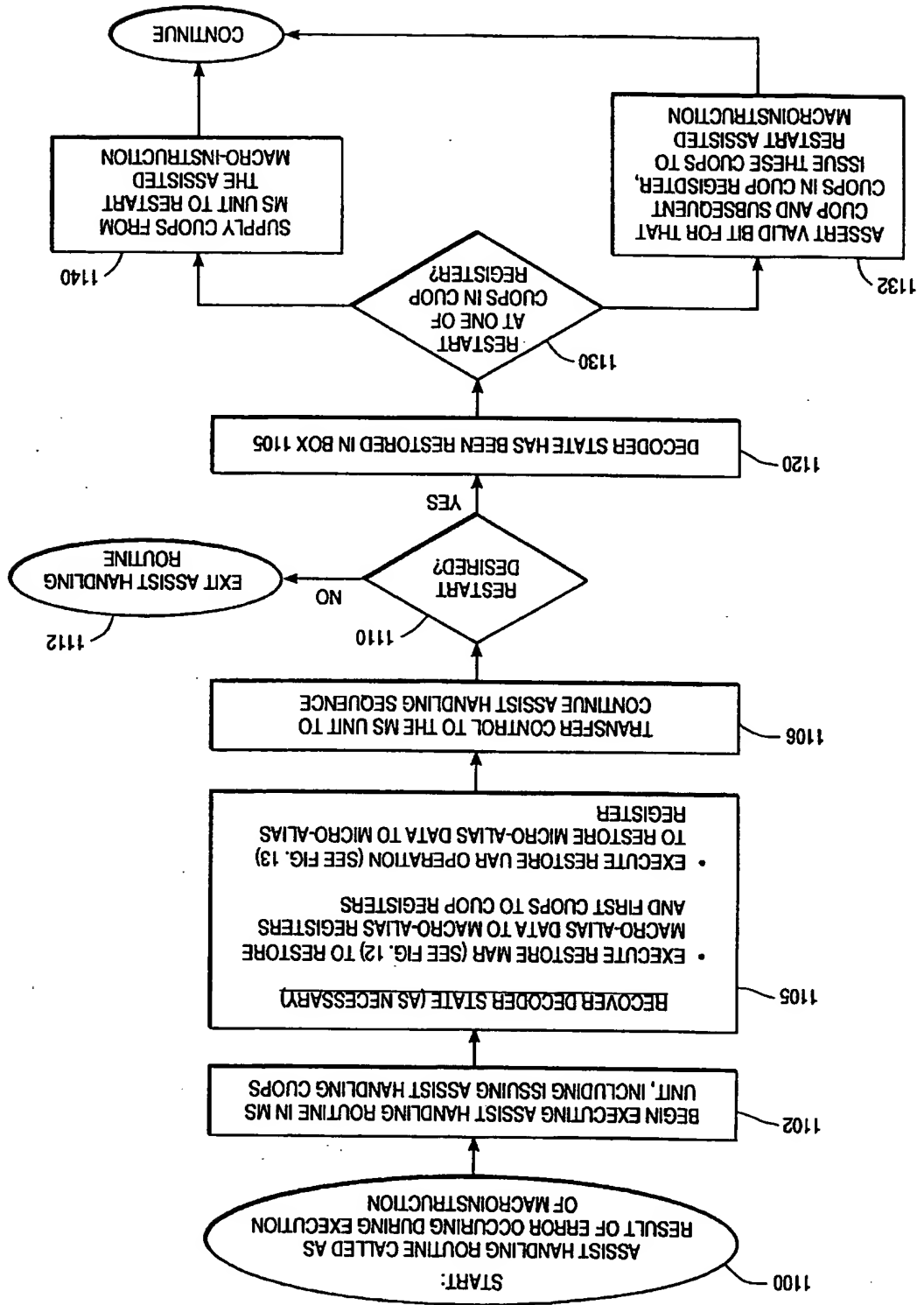


FIG. 11

	Docum ent ID	U	Title	Current OR
1	US 20020 11214 6 A1	<input type="checkbox"/>	Method and apparatus for synchronizing load operation	712/219
2	US 20010 01308 6 A1	<input checked="" type="checkbox"/>	Multiprocessor system and data transmitting method	711/119
3	US 65052 77 B1	<input checked="" type="checkbox"/>	Method for just-in-time delivery of load data by intervening caches	711/158
4	US 62634 06 B1	<input checked="" type="checkbox"/>	Parallel processor synchronization and coherency control method and system	711/141
5	US 60471 22 A	<input checked="" type="checkbox"/>	System for method for performing a context switch operation in a massively parallel computer system	709/108
6	US 57348 21 A	<input checked="" type="checkbox"/>	Method and apparatus for a direct data transmission between a communication network interface and a multimedia data processor	709/200
7	US 56299 50 A	<input checked="" type="checkbox"/>	Fault management scheme for a cache memory	714/805
8	US 55553 82 A	<input checked="" type="checkbox"/>	Intelligent snoopy bus arbiter	710/113
9	US 55532 66 A	<input checked="" type="checkbox"/>	Update vs. invalidate policy for a snoopy bus protocol	711/144
10	US 55532 58 A	<input checked="" type="checkbox"/>	Method and apparatus for forming an exchange address for a system with different size caches	711/3
11	US 53882 24 A	<input checked="" type="checkbox"/>	Processor identification mechanism for a multiprocessor system	710/104
12	US 53612 67 A	<input checked="" type="checkbox"/>	Scheme for error handling in a computer system	714/755
13	US 53197 66 A	<input checked="" type="checkbox"/>	Duplicate tag store for a processor having primary and backup cache memories in a multiprocessor computer system	711/146
14	US 47564 83 A	<input type="checkbox"/>	Jaw crusher with multiple drive means	241/101 .2

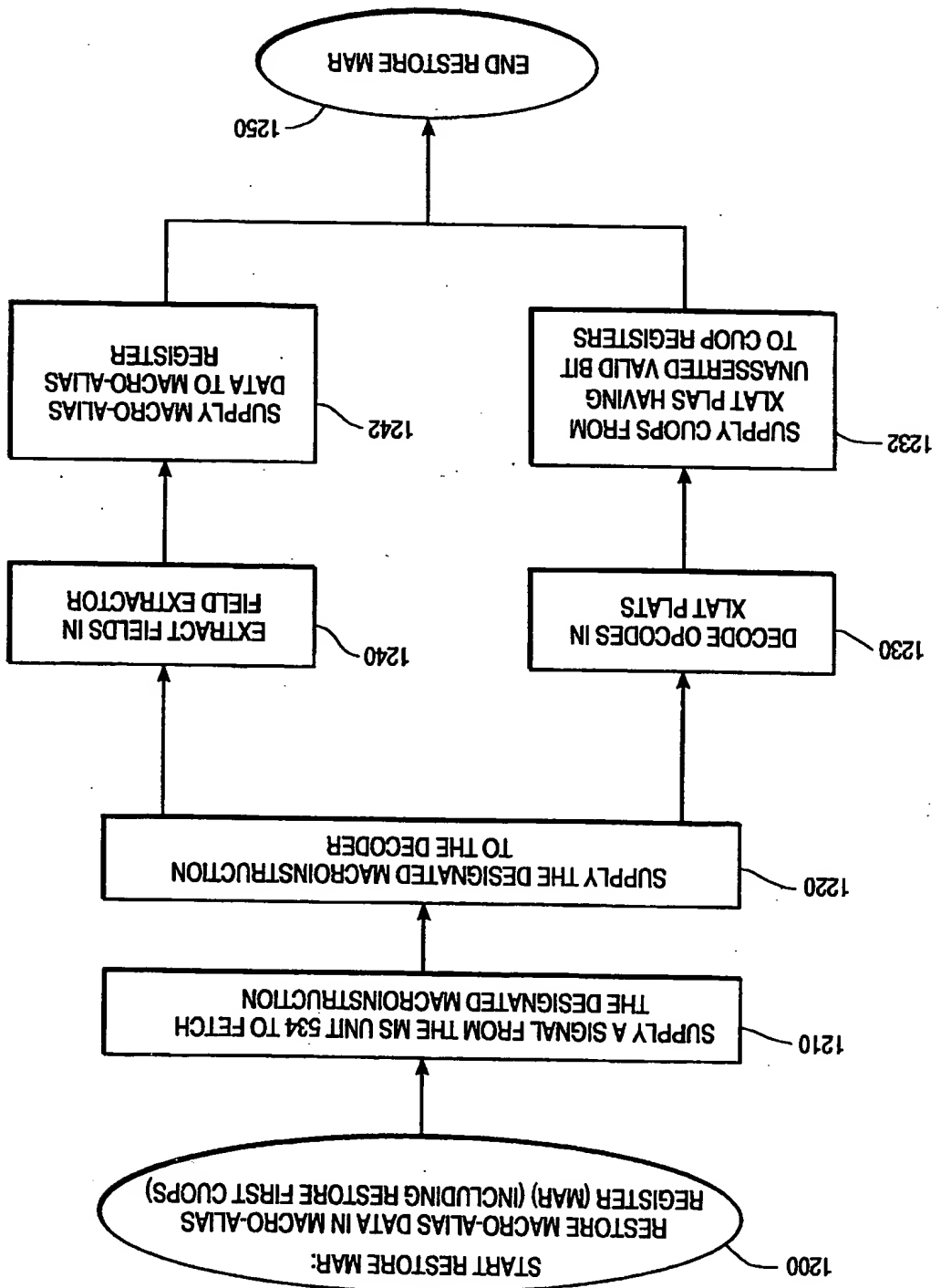


FIG. 12

	Docum ent ID	U	Title	Current OR
1	US 20020 11214 6 A1	<input type="checkbox"/>	Method and apparatus for synchronizing load operation	712/219
2	US 66067 02 B1	<input checked="" type="checkbox"/>	Multiprocessor speculation mechanism with imprecise recycling of storage operations	712/218
3	US 66009 59 B1	<input checked="" type="checkbox"/>	Method and apparatus for implementing microprocessor control logic using dynamic programmable logic arrays	700/7
4	US 55880 92 A	<input type="checkbox"/>	Printer control circuit and the printer controlled thereby	358/1.9

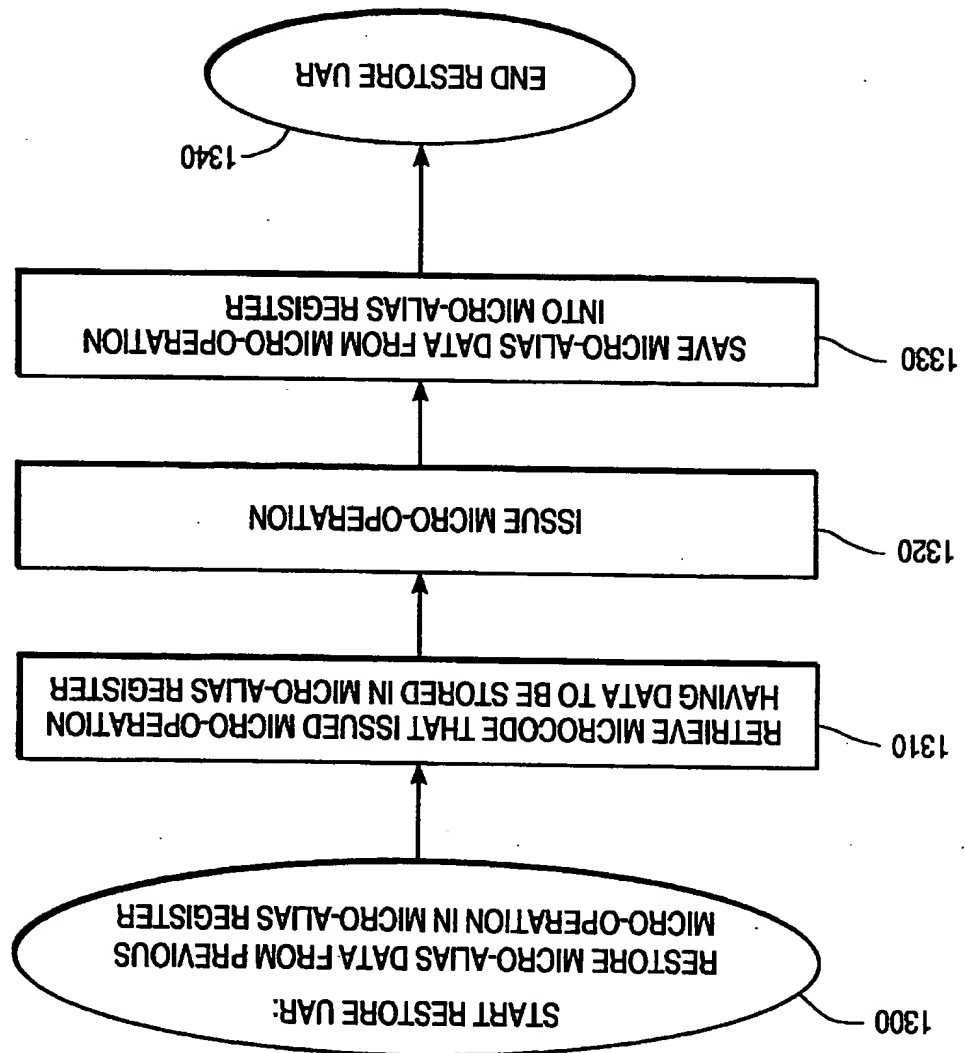


FIG. 13

	L #	Hits	Search Text	DBs
1	L1	1130	(fenc\$3 synch\$ barrier) near10 load near10 (instruction operation)	USPAT; US-PGPUB
2	L2	330	(glob\$4 coheren\$2 consisten\$2) and 1	USPAT; US-PGPUB
3	L4	311	(post pre) and 1	USPAT; US-PGPUB
4	L7	178	2 not (6 3 5)	USPAT; US-PGPUB
5	L6	143	2 and 4	USPAT; US-PGPUB
6	L3	14	(glob\$4 coheren\$2 consisten\$2) near99 1	USPAT; US-PGPUB
7	L5	4	(post pre) near99 1	USPAT; US-PGPUB
8	L8	344	(fenc\$3 synch\$ barrier) near10 load near10 (instruction operation)	EPO; JPO; DERWENT; IBM_TDB
9	L10	3	(glob\$4 coheren\$2 consisten\$2) and 8	EPO; JPO; DERWENT; IBM_TDB
10	L11	2	(post pre) and 8	EPO; JPO; DERWENT; IBM_TDB
11	L12	527	(fenc\$3 synch\$ barrier) near20 load near20 (instruction operation)	EPO; JPO; DERWENT; IBM_TDB
12	L13	4	(glob\$4 coheren\$2 consisten\$2) and 12	EPO; JPO; DERWENT; IBM_TDB
13	L14	4	(post pre) and 12	EPO; JPO; DERWENT; IBM_TDB
14	L15	173	(fenc\$3 synch\$ barrier) near10 load near10 (instruction operation)	EPO; DERWENT; IBM_TDB
15	L17	171	8 not 15	EPO; JPO; DERWENT; IBM_TDB

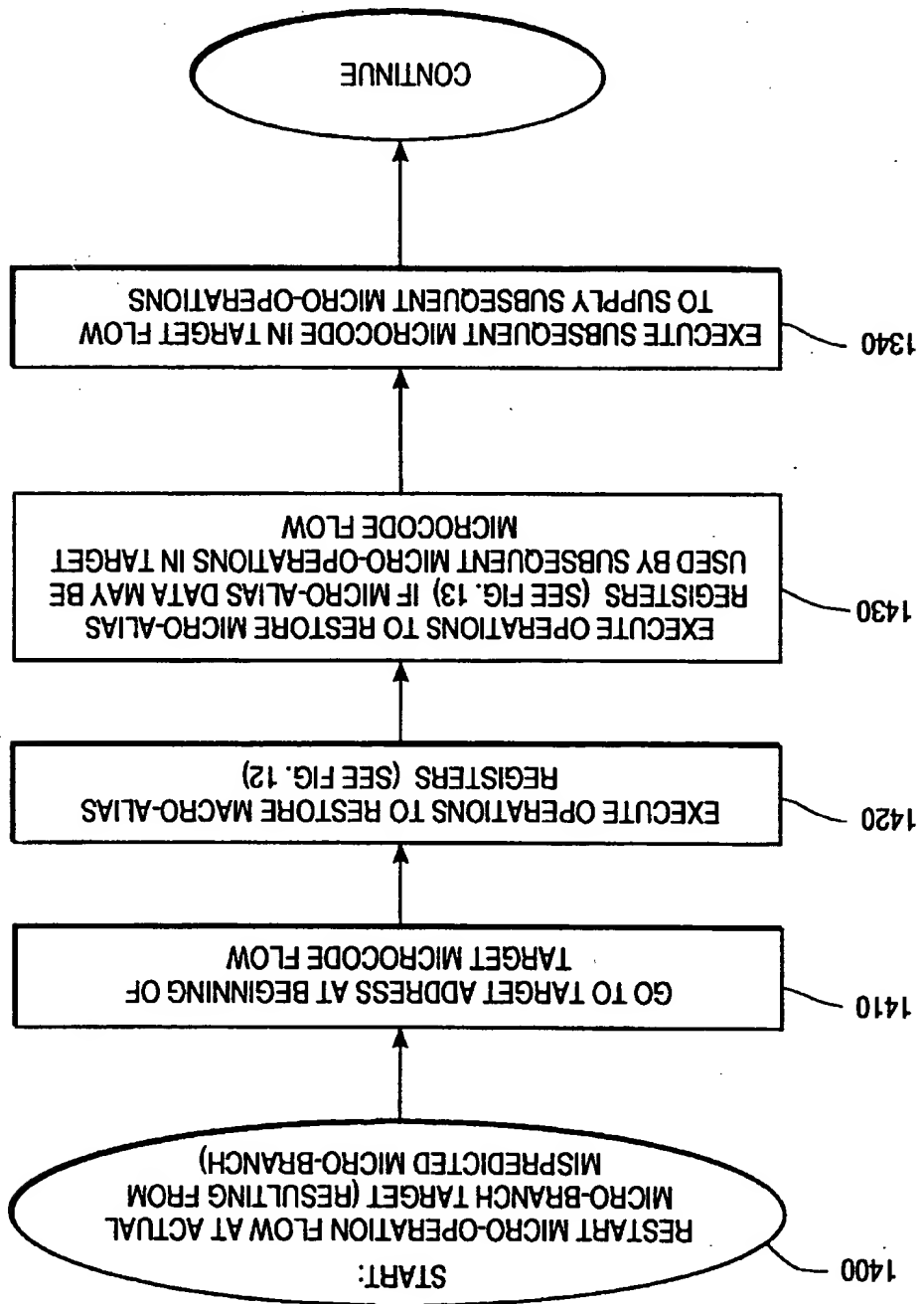


FIG. 14

	Docum ent ID	U	Title	Current OR
1	KR 20010 47533 A	<input type="checkbox"/>	Synchronous memory device	
2	US 57519 86 A	<input checked="" type="checkbox"/>	Computer system with self consistent ordering mechanism - cancels first LOAD operation and all subsequent operations based on comparison between address of STORE operation and address of first LOAD operation	
3	US 56110 70 A	<input checked="" type="checkbox"/>	Write-Load cache protocol for maintaining cache coherency and barrier synchronisation in multiprocessor computer system - updates remote cache in response to Write-Load instruction on bus from local processor when data with Write-Load instruction is stored in remote cache, without remote processor when data is not stored in remote cache	
4	EP 27215 7 A	<input type="checkbox"/>	Rock crushing system - has frame floatingly supporting jaws relative to frame structure, eccentric masses impart oscillatory vibration to one jaw	

DECODER HAVING INDEPENDENTLY LOADED MICRO-ALIAS AND SIMULTANEOUSLY BY ONE MICRO-OPERATION

This is a continuation of application Ser. No. 08/204,600,
filed Mar. 1, 1994 now abandoned.

CROSS-REFERENCE TO RELATED APPLICATIONS

Cross-reference is made to the following commonly
assigned copending patent applications: Ser. No. 08/204,

602, entitled "A Decoder for Decoding Multiple Instructions
in Parallel", filed Mar. 1, 1994, by Carbine et al.; Ser. No.

08/204,593, entitled "A Decoder for Single Cycle Decoding
Of Single Prefixes in Variable Length Instructions", filed

Mar. 1, 1994, by Brown, et al.; Ser. No. 08/204,744, "A
Method for State Recovery During Assist and Restart in a

Decoder Having an Alias Mechanism", Mar. 1, 1994, by
Boggs et al.; Ser. No. 08/204,862, entitled "A Method and

Apparatus for Aligning an Instruction Boundary in Variable
Length Macroinstructions with an Instruction Buffer", filed

Mar. 1, 1994, by Brown et al.; and Ser. No. 08/204,601,
entitled "A Method for Steering Multiple Variable Length

Instructions from an Instruction Buffer to Multiple Decod-
ers", filed Mar. 1, 1994, by Brown et al., now abandoned,

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to decoders having a micro-
code sequencing unit for supplying micro-operations during
long instruction flows. The decoder is useful to supply
multiple micro-operations in parallel.

2. Description of Related Art

Computers process information by executing a sequence
of instructions, which may be supplied from a computer
program written in a particular format and sequence
designed to direct the computer to operate a particular
sequence of operations. Most computer programs are written
in high level languages such as FORTRAN or "C" which are
not directly executable by the computer processor. In order
to run these high level programs, the program is compiled by
a compiler program that translates the higher level instruc-
tions into macroinstructions having a format that can be
decoded and executed. The compiled macroinstructions are
supplied to a decoder residing within the processor, where
each macroinstruction is decoded into one or more micro-
operations which are executable by the execution units in the
microprocessor.

Some decoders include aliasing mechanisms that extract
fields from a macroinstruction while simultaneously decod-
ing opcodes of that macroinstruction. An exemplary aliasing
system is described in U.S. Pat. No. 5,222,244, issued to
Carbine et al., on Jun. 22, 1993. One advantage of such
aliasing mechanisms is the availability of the extracted fields
in order to provide long micro-operation flows, some
decoders have included a microcode memory (ROM) and a
microcode sequencing unit. Particularly, the opcodes of the
macroinstruction generate an output from the PLA of the
decoder, which is then supplied to the microcode sequencing
unit in order to specify the entry point into the micro-

operation flow. Thereafter, a sequence of micro-operations
are provided from the microcode sequencing unit.

Microinstructions in such a flow have full access to all

extracted fields stored in the alias registers. However, in

some instances, additional information is needed: a micro-

operation in a long flow may need data fields from micro-

operations previously issued in the flow. Obtaining these

fields can be problematic because once issued from the

decoder, micro-operation fields can be difficult or impossible

to retrieve. In order to address this problem, one decoder

configuration that has been implemented in the 80960CF

microprocessor available from INTEL Corporation of Santa

Clara, Calif. utilizes a mechanism implemented by a micro-

code routine that stores data from a micro-operation only

sequentially, the stored data is routed back to one of the alias

registers in the decoder which data can only then be used by

subsequent microcode. Such a system requires a microcode

routine to implement storage of the issued micro-operation

in one of the alias registers. The system also consumes

expensive computer time, and is difficult to implement.

In a decoder having alias registers for storing extracted

alias fields, it would be an advantage to provide a configu-

ration for selectively storing information from a micro-

operation in a separate alias register before it is issued, so

that this information can be utilized by micro-operations

issued subsequently from the microcode sequencing unit.

Such a configuration would provide a straightforward way

of storing micro-operation fields for use by subsequent

micro-operations. Such a system could advantageously pro-

vide simultaneous availability of alias fields and micro-

operation fields at an alias multiplexer, in which they can be

utilized to assemble a complete micro-operation.

SUMMARY OF THE INVENTION

A decoder is disclosed that includes a micro-alias register
to store information from micro-operations for use by sub-
sequent micro-operations in the instruction flow. One advan-
tage of the described configuration is the ability for a Cuop
to simultaneously utilize micro-alias data and macro-alias
data to form an Auop in which all aliasing information has
been removed. Thus, a flexible approach for utilizing stored
alias fields to form a complete Auop is provided.

The decoder includes one or more translatable PLA ("XLAT

PLAs") and a microcode sequencing unit including a micro-

code ROM having micro-operation sequences stored

therein, and a microcode control circuit. A Cuop register is

coupled to the microcode sequencing unit and the XLAT

PLAs in order to store Cuops generated therefrom. A micro-

alias register is coupled to a multiplexer that is coupled to

the Cuop register to select data from one of the Cuops, which

is stored in said micro-alias register. The microcode

sequencing unit includes a control circuit, responsive to a

microcode instruction, to provide a control signal to select

one of the Cuops from which to store data into the micro-

alias register.

The capability to store information from a previous Cuop

provides a microcode programmer a straightforward way to

select a Cuop and store information therefrom that may be

useful for Cuops issued subsequently from microcode, with-

out requiring a lengthy routine to store the fields in a register

after the micro-operation has been completed and issued.

Thus, the apparatus and method described herein reduces the

length of microcode and therefore reduces the size and cost

of the microcode ROM.

	Document ID	U	Title	Current OR
1	JP 04191 204 A	<input type="checkbox"/>	AUTOMATIC WAREHOUSE	
2	JP 04036 672 A	<input checked="" type="checkbox"/>	ELECTRONIC COMPONENT TESTING INSTRUMENT	
3	DD 24755 4 A	<input checked="" type="checkbox"/>	Switching converter into operation - using starting thyristor across inverter and fired synchronously with rectifier thyristors	
4	US 41744 96 A	<input type="checkbox"/>	Monolithic solid state power controller - detects zero voltage crossing in source voltage and pre-selects mode synchronisation	

tradition only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention. The following description may include specific numbers and quantities associated with the apparatus and method described herein. For example, the processing apparatus and methods described herein can be practiced in a single microprocessor chip, or multiple chips, or in software. The chips may be made of silicon or other semiconductor material. Also, it should be apparent to one skilled in the art that the numbers and quantities utilized herein for illustrative purposes, for example the number of bits in a particular field can vary between embodiments.

The system described herein is particularly useful for decoding a sequence of variable length instructions that form a portion of a computer program. In the preferred implementation, the instructions supplied to the decoder (termed "macroinstructions") have the well-known format of the INTEL instruction set which is described, for example, in detail in the *i486™ Programmer's Reference Manual*, 1990, Section 2.4, available from INTEL Corporation. In this format, a macroinstruction can be up to fifteen bytes in length.

Each macroinstruction includes at least one opcode byte, which specifies the operation performed by the instruction. The length of an instruction is variable; i.e., the instructions have no set fixed length. For example, in one sequence of instructions, the first instruction may comprise twelve bytes, and the second instruction may comprise eight bytes, and the third instruction may comprise two bytes. There are certain format rules: for example, a macroinstruction is divided into sections that may include prefix bytes, followed by opcode bytes, which are followed by operand or other data bytes. The number, and even the existence of certain bytes is unknown with any certainty; the number of prefix bytes in a single macroinstruction may vary from zero to fourteen. The number of opcode bytes ranges from one to three, and the number of operand and other data bytes can vary from zero to eight. Prefixes, as used herein, are described with reference to the INTEL instruction set. Prefixes may include information relating to length of data, and control information regarding code segment override and data segment override. Furthermore, there may be one or more repeat prefixes, which indicate that an instruction is to be repeated a certain number of times. The *i486™ Programmer's Reference Manual*, 1990, Section 2.4, "instruction format", pages 2-15 and 2-16, includes a complete discussion of the instruction format of the INTEL instruction set, which includes various prefix bytes available. Also, section 26.2 "Instruction Format" on page 26-2 includes such a discussion.

The decoder described herein is designed to decode the INTEL instruction set, which includes instructions executable in the INTEL microprocessors including the 8086, 8087, 80286, i386™, 80287, i486™ and the Pentium™ microprocessors. It should be apparent however, that the structures described herein could also be used with instruction sets other than the INTEL instruction set. The following description includes circuits and methods suitable for implementation in a microprocessor. It should be recognized, however, that the principles described herein could be utilized in other processors.

Overview

Reference is first made to FIG. 1 which is an overview of a multiple instruction decoder of the preferred embodiment

BRIEF DESCRIPTION OF THE DRAWINGS

A technique is also disclosed that correctly stores the state of micro-alias registers in the event of a micro-branch misprediction or in the event of an error that requires restart at a location within a microcode sequence. This technique includes the steps of loading the macro-alias registers with fields extracted from the macroinstruction that accesses the microcode flow, and loading the micro-alias registers. This result is accomplished by writing microcode instructions LOADMAR and LOADUAR whenever the macro-alias registers and the micro-alias registers are required by microcode subsequent to a target that can be mispredicted or at which restart can occur.

FIG. 1 is an overview of a multiple instruction decoder having multiple decoders including a first decoder, a second decoder, and a third decoder, each of which receives a macroinstruction and decodes it into one or more micro-operations.

FIG. 2 is a block diagram of a partial decoder, such as the second decoder and third decoder, that can decode a subset of all executable macroinstructions.

FIG. 3 is a diagram of fields defined in the Cuop register. FIG. 4 is a block diagram of the field extractor and the macro-alias registers, illustrating the registers defined within the macro-alias register.

FIG. 5 is a block diagram of a full decoder, such as the first decoder that can decode all executable macroinstructions.

FIG. 6 is a diagram of a prefix decoding circuit for decoding each byte of an instruction buffer, and selecting one of said decoded prefixes, as appropriate, to supply to each decoder.

FIG. 7 is a flow chart of operations to decode the macroinstructions into a plurality of Auops.

FIG. 8 is a block diagram of circuitry including a cache memory for storing instructions, an instruction fetch unit, and a steering mechanism for steering instructions to multiple decoders.

FIG. 9 is a flow chart of operations to load the instruction buffer with one or more variable length macroinstructions, stored so that the first byte of a first macroinstruction is aligned with the first byte of the instruction buffer.

FIG. 10A is a flow chart that illustrates a procedure for steering multiple variable length instructions from the instruction buffer to multiple decoders.

FIG. 10B is a continuation of the flow chart of FIG. 10A. FIG. 10C is a continuation of the flow chart of FIG. 10B. FIG. 11 is a flow chart of operation of an assist handling routine, including, state recovery.

FIG. 12 is a flow chart of operations to restore a macro-alias data and the first Cuops supplied from XLAT PLAs in an instruction flow.

FIG. 13 is a flow chart illustrating operations to restore micro-alias data into a micro-alias register.

FIG. 14 is a flow chart illustrating operations to restart a micro-operation flow at an actual micro-branch target, such as may result from a mispredicted micro-branch.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIGS. 1 through 14 of the drawings disclose various embodiments of the present invention for purposes of illus-

	Docum ent ID	U	Title	Current OR
1	WO 20527 15 A1	<input type="checkbox"/>	SERVO MOTOR DRIVE CONTROL SYSTEM	
2	DE 19955 406 A1	<input checked="" type="checkbox"/>	Method for controlling multiple electrical loads operated from one source of power e.g. for motor vehicle, involves synchronizing time point and/or length of pulse triggering for operating current for loads with the operation of other loads	
3	EP 88493 2 A1	<input checked="" type="checkbox"/>	Ballast circuit with stabilised oscillator	
4	EP 82737 0 A2	<input checked="" type="checkbox"/>	Resonance type power converter unit, lighting apparatus for illumination using the same and method for control of the converter unit and lighting apparatus	
5	EP 70977 0 A2	<input checked="" type="checkbox"/>	Apparatus to control load/store instructions	
6	EP 69371 9 A1	<input checked="" type="checkbox"/>	Circuit for a household appliance	
7	EP 67999 0 A1	<input checked="" type="checkbox"/>	A computer apparatus having a means to force sequential instruction execution.	
8	EP 64571 5 A1	<input checked="" type="checkbox"/>	System and method for processing store instructions.	
9	EP 49363 4 A1	<input checked="" type="checkbox"/>	Electromagnetic valve control system.	
10	DE 39424 97 A1	<input checked="" type="checkbox"/>	Hydraulic load lifting device - has telescopic hydraulic cylinders designed so that inner and outer parts function simultaneously	
11	EP 42916 7 A1	<input checked="" type="checkbox"/>	Gear change mechanism.	
12	GB 22163 06 A	<input checked="" type="checkbox"/>	Load and synchronize computer architecture and process	
13	EP 31369 5 A1	<input checked="" type="checkbox"/>	Crane vessel.	
14	DE 37110 49 A1	<input checked="" type="checkbox"/>	Regulating device for at least two hydrostatic machines connected to a common operating-pressure line	
15	WO 88007 68 A1	<input checked="" type="checkbox"/>	DC TO DC CONVERTER CURRENT PUMP	
16	US 20030 20605 0 A	<input checked="" type="checkbox"/>	Synchronous load circuit operating apparatus includes frequency responder controlling frequency of clock signal to be above maximum operating frequency, if temperature is below maximum operating temperature	
17	WO 20030 81765 A	<input checked="" type="checkbox"/>	Synchronous reluctance motor control device has position/speed estimation/calculation unit for estimating position/speed of motor, based on predefined three phase voltage equation	
18	US 66256 60 B	<input checked="" type="checkbox"/>	Processor operating method in multiprocessor data processing systems, involves speculatively issuing load request to memory in response to load instruction, while barrier operations on interconnect are pending	
19	US 66091 92 B	<input checked="" type="checkbox"/>	Data processing system has controller at load/store unit to issue load request associated with load instruction which is output after barrier instructions to memory even before completion of barrier operation	
20	US 66067 02 B	<input checked="" type="checkbox"/>	Processor used in multiprocessor data processing system, discards data returned by memory when snoop invalidate affiliated with load request is received and load request is re-issued	

which is a part of a pipeline including other processing units 101, 10A, 10B and 10C. Briefly, the steering mechanism 101 is to be described in detail with reference to FIGS. 8, 9, 10A, 10B and 10C. Briefly, the steering mechanism 101 receives macroinstructions from an instruction buffer which is loaded by an instruction fetch unit that is coupled to memory or an instruction cache. For example, the macroinstructions may be fetched from memory or the instruction cache, and then stored in an instruction buffer for steering when needed by the decoders.

In the described embodiment, the macroinstructions include a first macroinstruction 102, a second macroinstruction 104, and a third macroinstruction 106. The macroinstructions 102-106 are provided in the sequence specified by the program structure. In other words, in the specified program sequence, the first macroinstruction 102 is followed by the second macroinstruction 104 which is followed by the third macroinstruction 106. In some instances, less than three macroinstructions may be available within the instruction buffer. In those instances, the first decoder receives the first available macroinstruction. If a second macroinstruction is available, the second decoder receives it, and then if a third macroinstruction is available, the third decoder receives it. Additional macroinstructions may be stored in the buffer, as is described elsewhere in more detail, and supplied to the decoders in subsequent cycles.

In the preferred implementation, no dependency checking is performed before the macroinstructions 102-106 is supplied to the parallel decoders, unlike other superscalar configurations which do not issue instructions in parallel if there are data dependencies between the instructions. Instruction scheduling is performed in units subsequent to the decoder.

The first macroinstruction 102 is supplied to a first decoder 110 that will be described in more detail subsequently. Briefly, the first decoder 110 is a "full" decoder that has the capability of decoding any macroinstruction within the instruction set defined for the microprocessor. The first decoder 110 includes circuitry to decode the first macroinstruction 102 into one or more micro-operations. As will be discussed subsequently in more detail, decoding of the first macroinstruction 102 proceeds several stages. First, the macroinstruction is decoded by an XLAT PLA to generate control micro-operations ("Cuops"). These Cuops are subsequently combined with extracted fields and immediate fields to supply Auops, which are illustrated at 112. The capital letter "A" represents "aliasing is resolved" (i.e., complete), and "uop" represents a micro-operation. Therefore, the first decoder 110 outputs Auops 112 in which indirect references have been resolved. The mechanism for aliasing will be described in more detail subsequently.

The second macroinstruction 104 is supplied to a second decoder 120, and the third decoder 130, compared with the first decoder 110, which is a full decoder. In the preferred embodiment, the second decoder 120 and the third decoder 130 have an identical structure, which is described in detail subsequently. The second decoder 120 produces a third Auop 132 from the first decoder 110, the second decoder 122 from the second decoder 120, and the third Auop 134 from the third decoder 130.

132 from the third decoder 130 are supplied to an output queue 140, which may be constructed in any conventional form. One embodiment for the output queue 140 is described in a copending, commonly assigned patent application entitled "An Output Queue for Multiple Decoders", by Brown et al Ser. No. 08/204,597, filed of even date herewith, which is incorporated by reference herein.

From the output queue 140, micro-operations 112, 122, and 132 are issued to other units for scheduling and eventual execution when those units are ready to receive the issued micro-operations 112, 122, 132. Auops are written into the output queue 140 independent of micro-operation reads from the queue 140. A micro-operation issued from the output queue 140 may be referred to as a "Duop", a term that represents a "decoder-issued Uop". A Duop may have a form identical with an Auop; however additional information may be included in a Duop such as a macro-branch target or a fall through address. Because the number of micro-operations varies widely from cycle to cycle, the output queue 140 holds a plurality of Auops, and issues multiple Duops when requested by subsequent units. In the preferred embodiment, the output queue 140 holds six micro-operations and can issue up to three micro-operations in parallel. The output queue 140 helps smooth the flow of micro-operations to subsequent units. Although not required for practicing the multiple instruction decoder described herein, the output queue 140 supplies the micro-operations to processing units that have a high execution rate. In some embodiments, these processing units include circuits that perform superscalar processing and/or out-of-order processing.

Discussion of Partial Decoder 200

Reference is now made to FIG. 2, which is a block diagram of a partial decoder 200 which, in the preferred embodiment, is implemented within the second decoder 120 and the third decoder 130. A macroinstruction 202, when supplied to the partial decoder 200, may or may not be decodable by the partial decoder 200. In order to make this determination, opcodes from the macroinstruction 202, which specify the operation to be performed by the macroinstruction, are provided to fast detection logic 210, which quickly analyzes them, and determines whether or not the partial decoder 200 can decode the macroinstruction 202. The fast detection logic 210 outputs a DECCAL ("Decodable By All") signal dependent upon the results of this determination. The DECCAL signal may be supplied to other units, including a steering unit that controls steering the macroinstructions 100 (FIG. 1). If the macroinstruction 202 is not decodable, it may be supplied in a subsequent Auop 112 to another decoder, for example the first decoder 110, which is a full decoder. In other words, if the macroinstruction 202 is nondecodable, it will be rescheduled to the next lowest available decoder during subsequent clocks until finally it reaches a decoder that can decode it. The full decoder, that can decode all macroinstructions in the instruction set, is the first decoder 110. Therefore, a macroinstruction 202 may be steered to the third decoder 130, and if not decodable therein, it can be re-steered to the second decoder 120 or to the first decoder 110.

The partial decoder 200, as well as the full decoder 500 is described, include an aliasing system. An exemplary aliasing system is described, for example, in U.S. Pat. No. 5,222,244, issued to Carbone et al. on Jun. 22, 1993.

The DECCAL signal together with all opcodes are provided to a translate (XLAT) PLA 220 within the partial decoder 200. "PLA" is an acronym for "Programmable Logic Array", which is a programmable circuit commonly

	Docum ent ID	U	Title	Current OR
21	US 20030 08425 9 A	<input checked="" type="checkbox"/>	Memory fence and load fence micro-architectural implementation method for speech synthesis, involves dispatching load fencing instruction to cache controller, after removing old loads from memory subsystem	
22	RU 22081 13 C	<input checked="" type="checkbox"/>	Method of automobile parking in multilevel mechanized parking lot, multilevel mechanized parking lot with keeping automobiles on trays, reception-rotary mechanism for multilevel mechanized parking lot with lifting cage	
23	JP 20032 12487 A	<input checked="" type="checkbox"/>	Forklift truck for cargo handling work has synchronization unit, which performs extrusion, pull-in, reverse and synchronous operations with advance operation of vehicle main body when detected tilt angle of mast is within suitable range	
24	JP 20030 61347 A	<input checked="" type="checkbox"/>	Switch-mode power supply changes load current supplied from main circuit, and maintains rectification circuit in synchronous rectification state or stop state	
25	JP 20030 28285 A	<input checked="" type="checkbox"/>	Automatic clutch control type vehicle controls connection/interruption of clutch, when power take-off mechanism is connected to transmission	
26	EP 12983 57 A	<input checked="" type="checkbox"/>	Electronic drive transmission control for synchro-shuttle or load switching transmission provides independent operation of switching elements of first and second transmission drive groups	
27	DE 20218 162 U	<input checked="" type="checkbox"/>	Piezoelectric transformer driver unit e.g. for fluorescent lamps, has pulse width modulation control providing signals of equal phase and frequency for synchronous operation of drivers, transformers, piezoelectric elements and loads	
28	DE 20218 163 U	<input checked="" type="checkbox"/>	Piezoelectric transformer-driver unit e.g. for computer LCD light-sources, uses driver unit to synchronously drive transformer units, piezoelectric units and loads	
29	RU 21994 63 C	<input checked="" type="checkbox"/>	Cross-country vehicle	
30	US 64502 98 B	<input checked="" type="checkbox"/>	Communication control system for elevators, adjusts timing for sending synchronization start data from specific slave to subsequent slave in order, based on specific timing criteria	
31	US 20020 11212 2 A	<input checked="" type="checkbox"/>	Cumulative ordering verification method in multiprocessor data processing system, involves determining whether load memory instruction executed after memory barrier instruction in remaining cache line is identified	
32	US 20020 11214 6 A	<input checked="" type="checkbox"/>	Load operation synchronization apparatus for computer system, has execution unit that executes decoded load fence instruction	
33	JP 20022 23434 A	<input checked="" type="checkbox"/>	Video monitoring system includes video server to transmit video data from camera along with synchronizing signal instruction, for controlling timing of synchronizing signal produced by cameras	
34	US 20020 08115 5 A	<input checked="" type="checkbox"/>	Cable pulling apparatus for use in e.g. pipe pulling operation, has articulating wedge of specified length, to grip or release cable depending on movement direction of adaptor	
35	JP 20020 78949 A	<input checked="" type="checkbox"/>	Pachinko machine has CPU to perform display control of LCD panel and motion control of character presentation mechanism	
36	JP 20020 21950 A	<input checked="" type="checkbox"/>	Speed change gear device for vehicle, has controller that operates main shaft rotation suppressing mechanism in predetermined condition during downshift operation of transmission	
37	JP 20020 10474 A	<input checked="" type="checkbox"/>	Power supply protection device using DC-DC converter, stops operation of DC-DC converter, when abnormal signal from voltage and load current monitors are input to synchronism monitor within prescribed time	
38	WO 20018 3869 A	<input checked="" type="checkbox"/>	Embroidery frame operation of embroidering machine involves synchronizing deceleration of accelerated embroidery frame and application of load to the frame	

used for decoders. The PLAs could be implemented as combinational logic, static or dynamic. In the preferred embodiment, the XLAT PLA 220 is implemented as static combinational logic. Within the XLAT PLA 220, the opcodes are decoded into a control micro-operation, termed a "Cuop" illustrated at 222. As implemented, a Cuop includes a template for a particular micro-operation and control information that defines aliasing (indirect data access) required by that Cuop. It may be noted that in the INTEL instruction set the number of opcodes bytes may be one, two, or three (the third byte is actually the mod/rm byte, which is considered by some to be an opcode byte). Predetermined bits from these three bytes are always supplied to the XLAT PLA 220. If during decoding some of the supplied bits are not necessary to decode the current instruction, then they are ignored. For example if only the first byte is an opcode byte, then the second and third bytes are ignored or considered "don't cares" by the XLAT PLA 220. The XLAT PLA 220 is also supplied with decoded prefix information. For example, the XLAT PLA 220 may be supplied with repeat prefix information, a lock indicator, and an operand size prefix.

The Cuop 222 is supplied to a Cuop register 224, in which it is latched by a latch clock signal. The Cuop 222 includes control fields 226 and a template 228, which are also illustrated in FIG. 3 and described subsequently with reference thereto. In the described embodiment of the partial decoder 200, the XLAT PLA 220 includes circuitry to produce one Cuop 222 per macroinstruction. Because there is only one XLAT PLA 220, the partial decoder 200 can decode only those macroinstructions 202 that can be decoded into one Cuop 222. However, in other embodiments, and particularly if more space is available, second and additional XLAT PLAs (not shown) could be provided to include circuitry to produce additional Cuops, so that two or more Cuops could be supplied in parallel.

Reference is now made to FIG. 3, which illustrates the fields defined by the Cuop 222 within the Cuop register 224. All Cuop data is stored in the Cuop register 224. Within the register 224 will not be issued unless a valid bit 310 is asserted.

The valid bit 310 is a 1-bit field that indicates whether or not the Cuop to which it is attached is valid. The valid bit 310 is particularly useful for a full decoder 500 to be described, which includes multiple XLAT PLAs each producing a Cuop, and a microcode sequencing unit. Specifically, the valid bit 310 is useful for determining whether a particular Cuop is valid before forming an Auop.

The control field 226 within each Cuop is useful for indirect access to other registers using an alias mechanism such as that to be described. The control field 226 may include opcode alias bits, which signify whether the opcode is alias. For example, the opcode alias bits may tell whether or not to substitute bits from alias registers or from other places. Furthermore, they may indicate whether or not the opcode fields need to be decoded to determine whether control bits can be replaced. One advantage of including control fields within each Cuop 224 is flexibility. This provides a substantial advantage because, by use of alias registers, and alias Cuops, the number of microterms (i.e., entries) in the XLAT PLAs can be reduced substantially. For example, in the preferred INTEL implementation one group of instructions includes the following operations: ADD, XOR, SUB, AND, and OR, which are implemented with one generic Cuop. Another group of instructions representable by only one Cuop includes ADC and SBB. The use of aliasing within the Cuops allows a single Cuop to be used for

each group of operations. An example of how a generic opcode can be utilized follows. Three bits, termed "TTT bits", are included within the first byte of certain opcodes. These bits are well known opcode bits of certain operations, and are shown in the i486™ *Programmers Reference Manual*, 1992, pages E-4 through E-6. The TTT bits are extracted by the field extractor (to be described) and included within the macro-alias data. A generic Cuop is formed without the TTT bits and then merged with the TTT bits from the macro-alias data within the alias multipliers to form a unique Auop.

The Cuop register 224 also includes an immediate field 350. In some instances, the immediate field 350 can specify one of the 32-bit alias registers, to be described. In other instances, the immediate field 350 can be used to address a constant stored in a ROM that stores commonly used constants. For example, the immediate field 350 may include a representation of an integer constant 65520, which is stored in a constant ROM that is addressed by the immediate field 350.

In summary, the control aliasing information provided in each Cuop is useful for reducing decoder size, and improving performance while still providing the capability for decoding a large and complex instruction set.

Referring again to FIG. 2, the first eight bytes from the macroinstruction 202, beginning with the opcodes, are supplied to a field extractor 250. These eight bytes will be supplied regardless of the actual length of the instruction. If, in a variable length instruction set, the actual number of bytes beginning with the opcodes is less than eight bytes, then the remaining bytes are still supplied to the field extractor 250, but they will not affect the microcode for that instruction. For example, if the length beginning with the opcodes to the end of the macroinstruction is five bytes, then the last three bytes of the eight supplied bytes may affect some of the fields of the macro-alias registers, but these fields will never be used. However, in the event that the length of the instruction beginning with the opcodes is greater than eight bytes, then this circumstance will be detected in the fast detection logic 210, and the instruction will not be decodable by the partial decoder 200.

The field extractor 250 includes circuits, which will be described subsequently in more detail, to extract fields such as register specifiers from the operands and steer them to alias registers. The field extractor 250 is responsible for extracting all aliasable information from the macroinstruction 202 in parallel with operation of the XLAT PLA 220. Preferably, the field extractor 250 could also include circuits that perform the following functions: detection of illegal operand usage, decoding of prefix bytes (if not previously decoded), and calculation of actual macroinstruction length for subsequent use. However, in one implementation, these functions are performed in the instruction steering logic.

The field extractor 250 receives, in addition to the opcodes and operands from the macroinstruction 202, control information 254 indicative of the mode of operation and how data is to be interpreted, such as a D-bit, which indicates whether data is interpreted to be 32-bit or 16-bit.

8

	Docum ent ID	U	Title	Current OR
39	KR 20010 54730 A	<input checked="" type="checkbox"/>	Parallel operation synchronizing circuit of uninterruptible power equipment	
40	WO 20014 5242 A	<input checked="" type="checkbox"/>	Method for controlling an high power AC-AC power converters, uses monitoring of load current to drive microprocessor which determines control signals for two double-controlled two-way semiconductor switches	
41	KR 20010 47533 A	<input checked="" type="checkbox"/>	Synchronous memory device	
42	FR 28023 60 A	<input checked="" type="checkbox"/>	Dimmer switch for lighting, comprises chopping transistors and microcomputer which enable a lighting load to be run at low voltage and voltage peaks to be counted to determine mode of operation	
43	DE 19955 406 A	<input checked="" type="checkbox"/>	Method for controlling multiple electrical loads operated from one source of power e.g. for motor vehicle, involves synchronizing time point and/or length of pulse triggering for operating current for loads with the operation of other loads	
44	JP 20011 37538 A	<input checked="" type="checkbox"/>	Image generation program recording medium for computer game apparatus, executes image displayed by frame display corresponding to adjusted synchronous time	
45	JP 20011 06482 A	<input checked="" type="checkbox"/>	Synchronization operating procedure for use in cranes, involves correcting number of revolutions of hydraulic motors based on operating characteristics of each hydraulic motor	
46	US 63447 25 B	<input checked="" type="checkbox"/>	Synchronous motor control device using vector control	
47	US 61788 67 B	<input checked="" type="checkbox"/>	Independent actuators arrangement for hydraulic or pneumatic control system, has actuators mechanically coupled to gear systems to monitor and provide indication of dissimilarity between operating motions of actuator	
48	JP 20010 02207 A	<input checked="" type="checkbox"/>	Load transfer device for fork apparatus has common controller which performs synchronous operation control of linear motors for movable forks by synchronous control of inverters connected to linear motors	
49	JP 20003 54399 A	<input checked="" type="checkbox"/>	Excitation control for synchronous machine, involves switching converter from AC excitation of frequency proportional to adjustable speed range of synchronous machine to low frequency excitation, during load cut-off	
50	CN 12742 11 A	<input checked="" type="checkbox"/>	Supervisory system for a base station in a communications system has supervised circuit cards each with a memory storing failure data and two supervisory cards one being active to read the failure information through a bus	
51	WO 20006 3775 A	<input checked="" type="checkbox"/>	Instruction scheduling method for processor of computer, involves hoisting instruction to earlier position in instruction list depending on barrier instructions	
52	JP 20002 03794 A	<input checked="" type="checkbox"/>	Synchronization controller for hydraulic actuator of winch apparatus has controller which drives two hydraulic actuators so that velocity detected by velocity sensor is equal to target	
53	JP 20001 84788 A	<input checked="" type="checkbox"/>	Synchronous motor controller controls current supply to field pole of each motor by acquiring phase difference of induced voltage and by detecting phase shift of current through field pole of motors	
54	JP 20000 12241 A	<input checked="" type="checkbox"/>	Remote monitoring and control system for fluorescent lamp lighting fixtures - controls voltage level in light control signal based on monitoring data to control switch operation for illumination control	
55	JP 11239 571 A	<input checked="" type="checkbox"/>	Scanned image synchronization device in magnetic resonance imaging apparatus for medical diagnosis - includes reflector to display scanned time synchronized ECG images which are penetrated based on predetermined synchronized ECG wave	

and the B-bit, which indicates stack size. Furthermore, the field extractor 250 may also receive a prefix vector 256 which includes decoded prefix information that can affect interpretation of the opcodes and operands. The Intel instruction set allows use of prefixes, which are bytes of information that may be prepended to a macroinstruction. One difficulty with the prefixes in the Intel instruction set is that any arbitrary instruction may or may not have a prefix, and if present, the number of prefix bytes can vary from one to fourteen. Generally, prefixes can affect the instruction code following it, particularly the interpretation of a macroinstruction 202 and the operands within it. Prefix decode logic, which will be described with reference to FIG. 6, decodes each byte in the instruction buffer to supply prefix vectors. For some purposes, the prefix decoding circuit 15 could be considered to be an element of the field extractor 250; however, for purposes of the present description, the prefix decoding circuit will be described separately. Briefly, the prefix decoding circuit decodes the prefixes to provide the prefix vector 256 having decoded bits easily usable by the field extractor 250 and the XLAT PLA 220.

The aliasable information, termed macro-alias data illustrated at 258, is stored in macro-alias registers 260. A latch clock signal is provided to latch the information therein for later use. It may be noted that the total data width of the operands and opcodes input into the field extractor 250 is sixty-four bits (8x8) for the partial decoder 200, but a large number of bits (about 130) of macro-alias data 258 are provided to the macro-alias registers 260. Therefore, it should be apparent that one macroinstruction 202 will not contain every alias field that can be extracted from the field extractor 250.

Reference is now made to FIG. 4, which is a block diagram of the field extractor 250 and the macro-alias registers 260, particularly illustrating fields within the macro-alias registers 260. The width of each register within the macro-alias registers 260 is dependent upon the maximum allowable size of the data stored within that register. For example, if the maximum size of source data is N-bits, then the size of the source register will be N-bits. Displacement data to be stored in another register may have a maximum size of 32-bits, and therefore the displacement register will have that size.

As discussed above, all aliasable information is latched into alias registers 260. Each field within the alias registers 260 falls into one of two general categories: bits extracted from the macroinstruction stream; or encoded fields that are a result of extracted data, architectural machine state, or some combination of the two. Examples of alias fields include logical registers, address size, data size, stack address and stack data size, immediate and displacement data, branch information, and portions of various predetermined opcodes.

The macro-alias registers 260 include fields such as a register segment field 270, a register base field 272, a register index field 274, an immediate data register 276, a displacement data register 278, a stack address size field 280, a data size field 284 which indicates the size of the data as well as some floating point information, and an address size field 286 that indicates whether the data address is 16- or 32-bits.

Reference is again made to FIG. 2. The macro-alias fields from the macro-alias registers 260 are provided to alias multiplexers 280, in which they are selectively combined with the Cuop 222 in order to assemble a complete Auop 290 in accordance with a predetermined aliasing system. Specifically, control fields that are included in the Cuop register 224 select macro-alias fields from the macro-alias registers 260, and combine them with a Cuop template from the Cuop register 224. The alias multiplexers 280 include conventional circuitry, including a plurality of multiplexers responsive to the control fields 226, that select particular fields and combine them with the template 228 from the Cuop register 224 in order to provide a complete Auop 290.

Discussion of Full Decoder 500

Reference is now made to FIG. 5, which is a circuit and block diagram of a full decoder 500 that can decode all macroinstructions. In comparison, the partial decoder 200 (FIG. 2) can decode only a subset of the macroinstructions. In the preferred embodiment, the full decoder 500 implements the first decoder 110 shown in FIG. 1.

A macroinstruction 502 having a format of the Intel instruction set is provided to the full decoder 500, in a manner discussed with reference to FIG. 1. Therefore, the macroinstruction 502 includes one or more opcode bytes illustrated at 504, and also may include one or more operands and other bytes following the opcodes. As discussed also with reference to FIG. 2, three bytes, beginning at the first opcode byte are supplied regardless of the actual number of opcode bytes, which may be one, two, or three. Additional bytes are included, for example if only the first byte is an opcode byte, then the second and third bytes may be ignored by the XLAT PLAs 510-516 and the entry point PLA 530.

The opcode bytes 504 are supplied to a first XLAT PLA 510, a second XLAT PLA 512, a third XLAT PLA 514, and a fourth XLAT PLA 516 that operate in parallel to supply up to four Cuops simultaneously from a single macroinstruction. Each XLAT PLA 510-516 receives the opcode bytes from the macroinstruction 502, as well as some prefix information, such as that described with reference to the partial decoder 200. The prefix information, supplied by the prefix decoder 200, is described subsequently in detail, is sometimes needed to interpret the opcode bytes. Each XLAT PLA 510-516 includes conventional circuitry to receive the opcodes 504 and responsive thereto to output a Cuop having the format described with reference to FIG. 3, including control fields and a template. In other words, the first XLAT PLA 510 includes circuitry to decode the opcode bytes into a first Cuop, the second XLAT PLA 512 includes circuitry to decode the opcode bytes into a second Cuop, the third XLAT PLA 514 includes circuitry to decode the opcode bytes into a third Cuop, and the fourth XLAT PLA 516 includes circuitry to decode the opcode bytes into a fourth Cuop. Thus, the XLAT PLAs 510-516 decode the opcode bytes of a macroinstruction 502 to supply up to four Cuops in parallel.

The full decoder 500 can decode macroinstructions into one, two, three, or four Cuops. In one embodiment, the valid bit 310 (see FIG. 3) is asserted in each valid Cuop by the XLAT PLA that produced it in order to signify that that Cuop is valid (unless there is an outstanding request to load the macro-alias registers, to be described). If, however, the Cuop is not valid, then the valid bit 310 for that Cuop is not asserted. If the macroinstruction 502 requires decoding into only one Cuop, then a Cuop output will be supplied from one of the PLAs 510-516 and its valid bit 310 will be asserted and the valid bits from the other XLAT PLAs will not be asserted. If the macroinstruction 502 is decoded into two Cuops, then the first Cuop and the second Cuop are supplied from two of the XLAT PLAs 510-516 and their valid bits 310 will be asserted. If the macroinstruction 502 is decoded into three Cuops, then the first three Cuops are supplied from three of the XLAT PLAs 510-516 and their valid bits 310 will be asserted. If the macroinstruction 502 is decoded into four Cuops, then the first four Cuops are supplied from four of the XLAT PLAs 510-516 and their valid bits 310 will be asserted.

	Docum ent ID	U	Title	Current OR
56	US 59432 46 A	<input checked="" type="checkbox"/>	Main AC power supply switching method for uninterruptable power supply	
57	JP 11047 937 A	<input checked="" type="checkbox"/>	Arc welding apparatus - has photoelectric unit for converting arc light generated between electrode and base material into electricity which is used for driving ventilating fan	
58	EP 88493 2 A	<input checked="" type="checkbox"/>	Ballast circuit for discharge tube e.g. fluorescent lamp - comprises applying AC voltage to resonant circuit, switched by two bridge-connected power semiconductors, and so supplying AC current to discharge tube connected to resonant circuit	
59	US 58327 77 A	<input checked="" type="checkbox"/>	Electromechanical transmission control apparatus - has number of electric motors for shifting gears of transmission, with load on synchroniser being measured by sensor when shifting gears	
60	JP 10136 682 A	<input checked="" type="checkbox"/>	Torque constant synchronous drive control apparatus for e.g. automatic lathe with material-supply machine - has torque control unit which adds component to setting torque command depending on detected acceleration, and regulates torque of torque control shaft	
61	JP 10121 454 A	<input checked="" type="checkbox"/>	Intrusion test device e.g. Swedish type sounding test device for measuring ground hardness and strength - has weight supply mechanism which supplies heavy weight to weight placing stand of load piler elevated by hydraulic cylinder for rod lift to predetermined height	
62	JP 10121 453 A	<input checked="" type="checkbox"/>	Intrusion test device e.g. Swedish type sounding test device for measuring ground hardness and strength - has measuring unit which measures amount of intrusion of rod for intrusion test in apparatus body	
63	US 57519 86 A	<input checked="" type="checkbox"/>	Computer system with self consistent ordering mechanism - cancels first LOAD operation and all subsequent operations based on comparison between address of STORE operation and address of first LOAD operation	
64	JP 33175 03 B	<input checked="" type="checkbox"/>	Operation synchronising method for service control points in load sharing group - generating new control by any one SCP in group, updating list at any one SCP corresponding to itself to add new control, sending add control signal identifying new control to all other SCPs in group to update their control list	
65	JP 10094 290 A	<input checked="" type="checkbox"/>	Load control apparatus e.g. for motor - generates common power supply synchronising with operation of thyristor and varies input impedance of control circuit	
66	EP 82737 0 A	<input checked="" type="checkbox"/>	Resonance-type power converter e.g. for fluorescent and electrodeless lamps - uses NAND drive circuits to operate power MOSFETs in accordance with voltage detecting comparator outputs	
67	JP 10026 026 A	<input checked="" type="checkbox"/>	Spark ignition IC engine - has controller which operates two inlet valves to open suction ports synchronously during high load operation and closes one of ports with valve while allowing other port to open during low load	
68	JP 09331 680 A	<input checked="" type="checkbox"/>	Load drive circuit for e.g. fluorescent lamp, liquid crystal back light - synchronises operations of switching transistor and inverter with signal output to switching transistor as chopping drive signal, based on comparison result between AC waveform of inverter and feedback signal	
69	JP 09255 298 A	<input checked="" type="checkbox"/>	Clamping apparatus for rolled article e.g. machine glazed paper e.g. newspaper loaded on fork lift - has split arms whose opening and closing operations are synchronised by frictional force of lining pad provided between attachment shafts, at no load condition	
70	JP 09252 541 A	<input checked="" type="checkbox"/>	Individual load operation detector for synchronous generator of power supply distribution system - has reactive power adjustment unit which is connected to output end of synchronous generator	
71	JP 09247 946 A	<input checked="" type="checkbox"/>	Inverter for DC to AC conversion and load current control e.g. for AC synchronous motor - has switching unit into which required ratio, obtained from operation unit, is fed as correction signal to correct ripple voltage included in DC output	

will be asserted. And of course, if the macroinstruction 502 is decoded into four Cuops, then the four XLAT PLAs 510-516 supply the four Cuops in parallel, each having its valid bit 310 asserted.

In order to handle the situation in which the macroinstruction 502 is to be decoded into multiple Cuops that are not all supplied from the XLAT PLAs 510-516, the full decoder 500 includes circuitry for accessing microcode, including an entry point PLA 530 and a microcode sequence ("MS") unit 534 coupled thereto that includes control circuitry to perform the following described functions. The opcodes from the macroinstruction 502 are supplied to the entry point PLA 530 that decodes the opcodes to generate an entry point address into microcode ROM. The generated entry point address is supplied to the MS unit 534 that, responsive to the entry point, generates a series of Cuops. The MS unit 534 is capable of generating multiple UROM Cuops in any one cycle, responsive to the entry point generated in the entry point PLA 530. As implemented, the MS unit 534 generates up to three UROM Cuops per cycle. The UROM Cuops have a format similar to that of the Cuops generated in the XLAT PLAs 510; particularly, the UROM Cuops include control fields and a template.

The UROM Cuops are generated by microcode within the MS unit 534. The microcode can, in some instances, generate particular signals such as LOADMAR, LOADUAR, and other signals which will be discussed subsequently in more detail.

As implemented, the MS unit 534 allows branching operations, which are termed "micro-branches". In the event of a branch misprediction, which will be determined by state may need to be recovered before microcode at the actual target address can be executed. Micro-branch misprediction is discussed in more detail with reference to FIG. 13.

A Cuop multiplexer unit 540 includes a plurality of 2:1 multiplexers that select between a Cuop from the XLAT PLAs 510-516, and UROM Cuops from the MS unit 534. Specifically, the first 2:1 multiplexer selects between a Cuop from one of the first XLAT PLAs 510 and the MS unit 534. The second 2:1 multiplexer selects between a Cuop from the second XLAT PLA 512 and the MS unit 534, and the third 2:1 multiplexer selects between the third XLAT PLA 514 and the MS unit 534. Control for the plurality of 2:1 multiplexers 540 is supplied from the MS unit 534 via a MSBus signal. The MS unit 534 includes control logic, responsive to a valid entry point, to generate a MSBus signal that controls a multiplexer unit that selects between the Cuops from the XLAT PLAs 510-516 and UROM Cuops. When the MS unit 534 wants to begin inserting UROM Cuops, it asserts the MSBus signal to switch the multiplexers 540, and maintains it as long as microcode control is required. In other words, when the MS unit 534 begins supplying UROM Cuops, it asserts the MSBus signal in order to direct the multiplexers 540 to select the UROM Cuops instead of the Cuops from the XLAT PLAs 510-516.

In the preferred embodiment, some of the more frequently-used macroinstructions are decoded by the XLAT PLAs 510-516 into one, two, three, or four of the first Cuops in the micro-operation sequence, which provides high performance at the cost of additional minterms in the XLAT PLAs 510-516. Alternatively, for some less frequently-used macroinstructions, the four XLAT PLAs 510-516 issue no Cuops, but simply allow the MS unit 534 to issue all Cuops. This second alternative has a disadvantage of lower performance (i.e., a loss of at least one clock cycle), but can save minterms (entries) in the XLAT PLAs 510-516, which is a design trade-off that reduces die space at the expense of lower performance. This trade-off can be useful for less frequently used instructions or for long microcode flows where the significance of one additional clock is lessened. The Cuop from each of the 2:1 multiplexers 540 is latched into a Cuop register unit 550 that includes a plurality of Cuop registers. Specifically, a first Cuop register receives a first Cuop from the first 2:1 multiplexer, a second Cuop register receives a second Cuop from the second 2:1 multiplexer, a third Cuop register receives a third Cuop from the third 2:1 multiplexer, and a fourth Cuop register receives a fourth Cuop directly from the fourth XLAT PLA 516. The Cuops are latched into the Cuop register unit 550 by a latch clock signal that is supplied by conventional control means. The contents of each Cuop register include the fields described previously with reference to FIG. 3. Specifically, each Cuop includes a valid bit 310, control fields 226, and template fields 228. Only if its valid bit 310 is set will a Cuop be used to form an Auop that is issued into the output queue.

The contents of the four Cuop registers 550 are supplied to a 4:1 micro-alias multiplexer 560 that selects one of said Cuops, which is supplied to micro-alias registers 562 and stored therein. The information in the micro-alias registers 562 is utilized to hold information extracted from one of the four Cuops for use by a Cuop subsequently issued by the MS unit 534. The micro-alias register 562 includes three fields: an 8-bit SRC1 field, an 8-bit SRC2 field, and an 8-bit DEST. Control for the micro-alias register 562 is supplied from the MS unit 534. Particularly, the MS unit 534 supplies a LOADUAR ("Load Micro-Alias Register") signal that loads the micro-alias register 562, and a UARLUP ("Micro-Alias Register Micro-Instruction Pointer") signal that includes two bits to select one of the four Cuops from which to store information. The Cuop field stored in the micro-alias registers 562 can be from either the XLAT PLAs 510-516, or from the MS unit 534. The MS unit 534 supplies a UPMUROM signal that indicates whether the Cuop is to be loaded from the XLAT PLAs 510-516, or from the MS unit 534.

The micro-alias registers 562 are particularly useful in long instruction flows, which allows the microcode programmer flexibility in retaining information and simplifying code. For example, the microcode may include a rounding routine that is generally used in many different microcode sequences to round a number. Because the rounding sequence will likely be stored by a microcode instruction not within the rounding routine, the rounding sequence itself would not know directly where the information was stored. Therefore, the preceding microcode stores the address of this number (i.e., the register in which it is stored) in the micro-alias registers 562, and then the rounding routine is run, using the contents of the micro-alias register 562 to specify the register address of a number to be rounded. Thus, by storing the address of the register in which the data is stored, a common generic microcode routine can be utilized by any of a number of other microcode programs. Because the registers are not hard-coded into any routine, the programmer has great flexibility to access any register indirectly by accessing the register address within the micro-alias register 562. For example, in a microcode flow for a

	Docum ent ID	U	Title	Current OR
72	JP 09138 440 A	<input checked="" type="checkbox"/>	Opening and closing mechanism for barrier spring for photo-graph optical system - operates two barrier springs individually by two driving shafts to open and close opening in frame	
73	JP 09135 557 A	<input checked="" type="checkbox"/>	Brushless excitation appts of variable speed type synchronous motor - has non-contacting type transmitter which transmits field current from stator winding wire to semiconductor element in which current flow is controllable	
74	DE 29704 520 U	<input checked="" type="checkbox"/>	Power supply for gas discharge lamp - has controller keeping lamp current constant irrespective of load and reversing current synchronously with operation of shutter	
75	JP 09120 388 A	<input checked="" type="checkbox"/>	Information processing system with data synchronization function - has control unit which sends information on whether node is operating synchronously, to mediation unit	
76	JP 09068 737 A	<input checked="" type="checkbox"/>	Barrier opening-closing mechanism for protecting photograph optical system with lens - has spring provided between notches of barrier ring connecting individually with interlock pins of barrier vane which shuts or opens photograph opening of lens body tube frame	
77	US 56110 70 A	<input checked="" type="checkbox"/>	Write-Load cache protocol for maintaining cache coherency and barrier synchronisation in multiprocessor computer system - updates remote cache in response to Write-Load instruction on bus from local processor when data with Write-Load instruction is stored in remote cache, without remote processor when data is not stored in remote cache	
78	JP 09016 529 A	<input checked="" type="checkbox"/>	Information processing system - has number of information processors each one having CPU where synchronous operation is performed between them	
79	US 56871 39 A	<input checked="" type="checkbox"/>	Load control and management system for electrical power networks - uses recycle counter to open and close switch in control line of individual units of equipment whilst control signal is present.	
80	US 55663 07 A	<input checked="" type="checkbox"/>	Data processor contg pipelining system instruction executing - by sync writing value indicated by data load instruction into first stack pointer and second stack pointer when execution of data load instruction is completed	
81	EP 73136 1 A	<input checked="" type="checkbox"/>	Power performance test unit for electric vehicle motor - has test motor linked to load motor both having controller that are fed control data from host computer which measures data	
82	JP 08205 378 A	<input checked="" type="checkbox"/>	Load control device used in residence and building - includes 'OFF' control part which outputs opening instruction to switch based on contents of setting memory pertinent to current detector's output	
83	EP 70977 0 A	<input checked="" type="checkbox"/>	Pipelined processor for issuing and executing multiple instructions out of order every machine clock cycle - permits load and store instruction to issue and execute out of order and incorporates unique store barrier cache used to dynamically predict whether or not store violation is likely to occur	
84	JP 08103 100 A	<input checked="" type="checkbox"/>	Operation controller of AC excitation synchronous machine - controls excitation converter based on command such that power load angle is reduced	
85	GB 22936 76 A	<input checked="" type="checkbox"/>	Counter circuit having load function - has at least three counter circuits corresp. to numerical digit, with OR circuit receiving output from load value monitor circuits	
86	JP 08088 980 A	<input checked="" type="checkbox"/>	Independent operation detector for distributed power supply of solar power generation system - outputs independent operation detection signal and continues synchronisation with same timing, when detected phase change in load voltage exceeds set value	
87	RU 20566 98 C	<input checked="" type="checkbox"/>	Electric power source with non-transforming input - has comparators to form control signals for thyristors of converter and regulator and control connection of source to load	
88	EP 69371 9 A	<input checked="" type="checkbox"/>	Domestic washing machine electrical circuit - has control microprocessor for synchronised control and regulation of electrical loads with incorporated voltage monitoring.	

floating point routine such as sine or cosine, a temporary floating point value may be stored in a "scratch" register, whose address is stored in the micro-alias register 562. Then, the generic rounding routine may be called, which utilizes the address within the micro-alias register 562 to specify the value that is to be rounded for the floating point routine. Thus, one generic rounding routine that indirectly accesses the scratch register via the micro-alias register 562 is provided, which has an advantage of substantially reducing the amount of microcode, thereby reducing cost and die space.

The micro-alias register 562 is provided within the full decoder 500, and not in the partial decoder 200, because only the full decoder 500 includes the microcode sequencing unit that is capable of decoding the complicated instructions which can utilize the micro-alias register 562. As an additional advantage, the micro-alias register 562 can be aliased within themselves to provide two levels of indirection. For example, a Cuop may have its source aliased to use logical SRC1 in the micro-alias register 562, while logical SRC1 in the micro-alias register 562 is aliased to the contents of a macro-alias register 580, described subsequently. Therefore, the output Auop will receive the contents of its SRC1 field from the designated macro-alias register. In order to accomplish this, the micro-alias register also includes an alias bit for each field that indicates whether or not the micro-alias field is aliased to a macro-alias register. If not aliased, then the data within the micro-alias register can be used literally, but if it is aliased, then the data within the micro-alias register indirectly specifies one macro-alias field. To implement this, the alias bit can be used as a control input into the alias multiplexer.

Returning to the upper section of FIG. 5, the opcodes and operands from the macroinstruction 502 are supplied to a field extractor 570. Because the full decoder 500 can decode any of the macroinstructions in the instruction set, and the maximum length of a macroinstruction in the preferred instruction set is fifteen bytes, the field extractor 570 receives the entire eleven bytes that is the maximum size of the opcodes and operands of a macroinstruction. The field extractor 570 receives, in addition to opcodes and operands from the macroinstruction 502, control information 574 that, responsive to the mode of operation of the processor indicates how data is to be interpreted. An example of control information 574 is a D-bit, which indicates whether data is interpreted to be 32-bit or 16-bit, and the B-bit which indicates stack size. Furthermore, the field extractor 570 may also receive a prefix vector 576 that can affect interpretation of the opcodes and operands, if the instruction code allows prefixes that affect interpretation of the macroinstruction 502 and the operands within it. The prefix vector 576 is provided from a prefix decoding circuit described subsequently with reference to FIG. 6.

The output of the field extractor 570 includes macro-alias data 578 which is stored in macro-alias registers 580. The structure of the macro-alias registers 580 is identical to that described with reference to FIGS. 2 and 3. Each decoder, whether it be the full decoder 500 or the partial decoder 200, includes a field extractor that is responsible for extracting all aliasable information from a macroinstruction in parallel with operation of the XLAT PLAs in the full decoder. Particularly, the full decoder 500 includes only one field extractor 570 and one set of macro-alias registers 580, even though there are multiple (i.e., four) XLAT PLAs 510-516. By reference to FIG. 2, it may be noted that each partial decoder 200 also includes a field extractor and a set of macro-alias registers, although they may not be as large

as in the full decoder 500 and may not be able to support the full range of operands. Thus, an advantage is provided because only a single field extractor is required for multiple XLAT PLAs, each of which can independently supply a Cuop. Thus, multiple Cuops can be supplied in parallel.

A plurality of alias multiplexers 590 are each coupled to the macro-alias registers 580, the micro-alias registers 562, and the Cuop registers 550. Specifically, a first alias multiplexer 592 is coupled to the first Cuop register, the micro-alias register 562, and the macro-alias registers 580. The second alias multiplexer 594 is coupled to the second Cuop register, the micro-alias register 562, and the macro-alias registers 580. The third Cuop register, the micro-alias register 562, and the macro-alias registers 580, and the fourth alias multiplexer 596 is coupled to the fourth Cuop register, the micro-alias register 562, and the macro-alias registers 580. In the event that a micro-operation causes an error, there are some instances in which the error can be corrected by assist handling microcode within the M5 unit 534. In those instances, the M5 unit 534 is directed, by a fault or exception handler that has been notified of the error, to assist the instruction and, as necessary, to restart the instruction. Assist handling can only occur in the full decoder 500, because assist handling is directed by the M5 unit 534 which only has access to macro-data in the full decoder 500. Further discussion of assists and restarts, and the state recovery necessary to perform them is described in more detail with reference to FIGS. 11, 12, 13, and 14.

Reference is now made to FIG. 6, which is a diagram of a prefix decoding circuit that supplies prefix information to both the full decoder 500 and the partial decoder 200. As illustrated, the instruction buffer 600 includes sixteen contiguous bytes of instruction code, which have not been separated into individual macroinstructions. However, a plurality of byte marks 602 are provided by a predecoder, such as the instruction length decoder 814, discussed with reference to FIG. 8. The byte marks 602 can also be supplied by conventional means to designate the positions of the last byte of each instruction, and the first opcode byte of each instruction, thereby indicating a prefix by the number of bytes between the first operand byte and the last byte of the previous instruction. Additionally, steering control information is provided as illustrated at 604, which indicates which instruction is being provided to each of the decoders.

The byte marks 602 and the steering control information 604 are supplied to prefix control logic 610, in which the information is utilized to determine if a one-byte prefix is present and if so, to direct the decoded prefix information to the decoder that receives the associated opcodes. The prefix many prefixes exist in each instruction. If only one prefix is present, or none, then all prefix decoding can be handled within the prefix decoding circuit to be described. However, control logic 610 also includes circuitry to determine how many prefixes exist in each instruction. If only one prefix is present, or none, then all prefix decoding can be handled within the prefix decoding circuit to be described. However,

	Docum ent ID	U	Title	Current OR
89	JP 08009 693 A	<input checked="" type="checkbox"/>	AC synchroniser secondary excitation control method for pumping electric generator - by continuing operation without stopping converter for excitation even when rotational speed of AC excitation synchronous machine and reversible pump turbine increases due to load interruption	
90	JP 07297 685 A	<input checked="" type="checkbox"/>	Pseudo-random number sequence generator e.g. for spread spectrum application - uses pseudo noise mask conversion mechanism with predetermined conversion matrix and calculates pseudo noise mask data and converts comparison value respectively	
91	EP 67999 0 A	<input checked="" type="checkbox"/>	Digital computer providing forced sequential instruction execution - maintains process bit and memory attribute bit associated with shared memory page to order, store and load operation	
92	EP 67791 1 A	<input checked="" type="checkbox"/>	Islanding operation prevention appts. of dispersed power generation system for supplementing large scale power plant - comprises circuit breaker between dispersed generators and utility supply, power value detector, synchronous circuit, switch and controller	
93	JP 07253 143 A	<input checked="" type="checkbox"/>	Gearchange mechanism for car - arranges synchronisation and engagement mechanism in extension of sun gear and transmits power from planet gear to upstream side of power transmission route	
94	JP 07184 268 A	<input checked="" type="checkbox"/>	Operation terminal for remote supervisory control system using small data memory - has time multiplexed terminals, which supply operation data in signal return period synchronised with transmission signal, and generates control data for load NoAbstract	
95	US 54249 69 A	<input checked="" type="checkbox"/>	Product sum operation unit for high speed processor - has latch loading instruction on timing signal with second latch loading instruction synchronously with second signal to give control signal to adder unit	
96	JP 07107 542 A	<input checked="" type="checkbox"/>	Synchronised down load system for vehicle telephone - has main memory which returns to synchronised operation after transfer of load module under asynchronous operation	
97	EP 64571 5 A	<input checked="" type="checkbox"/>	Interface system for coupling main processor to bus controller - accepts processor synchronised program input-output operations and asynchronously manages execution of program I/O within I/O system	
98	JP 07039 161 A	<input checked="" type="checkbox"/>	Synchronised rectifier type switching power supply - uses main switching element in transformer primary MOSFET rectifier elements switched by bipolar transistors on secondary side and synchronises operation of switching and rectifier	
99	RU 20164 62 C	<input checked="" type="checkbox"/>	Synchronising a static frequency converter and AC mains - is by formation of a supplementary signal coinciding in frequency and phase with the output voltage of the static converter	
100	EP 60183 1 A	<input checked="" type="checkbox"/>	Electrical detonator load groups activation system for quarry blasting - has slave auxiliary control units for generating local control signals from master control signals, for initiating operation of electrical delay devices	
101	US 52671 20 A	<input checked="" type="checkbox"/>	Electromechanical relay control device for domestic appliance - operates relay to supply or remove power from load to preselected interval in AC waveforms and prevents decreasing area of content	
102	KR 93053 81 B	<input checked="" type="checkbox"/>	Controlling parallel operation of inverter - generating instantaneous voltage control value synchronised by common carrier, and compensating difference between reference of load current and output current NoAbstract	
103	RD 34612 2 A	<input checked="" type="checkbox"/>	Handling of load and store multiple operations with source and sink general point register synchronising - turning on inhibit tag CTL trigger field if operation utilises more than two GPR(s), turning field off after completion, and accepting identification tags	
104	SU 17745 29 A	<input checked="" type="checkbox"/>	Electric arc welding and surfacing device - has comparator to synchronise operation of pulse shapers, and uses resistor to shunt thyristor switches forming initial current	

if more than one prefix is associated with an instruction, then a conventional state machine (not shown) will consume one prefix each clock, until all prefixes have been decoded into a prefix vector. Of course, further decoding of the macroinstruction to which the multi-byte prefix is prepended will be stalled until the prefix vector becomes available after it has been fully decoded.

In the preferred embodiment, both the full decoder 500 and the partial decoder 200 can handle one prefix byte per macroinstruction without a clock cycle penalty. In the preferred configuration, prefixes are not considered a separate instruction, but are treated as another byte of the instruction to which they are prepended. The assumption of one prefix byte per macroinstruction is particularly advantageous if it turns out that instruction does have only one prefix byte.

As implemented, the partial decoder 200 can handle only one prefix byte per macroinstruction. The full decoder 500 can handle the rare instance of multiple prefix bytes. If an instruction were to be stored to a partial decoder 200 that has more than one prefix byte, then the steering mechanism will invalidate the decoding of that instruction and subsequent operations based on that instruction, and restore it to the lowest available decoder during subsequent clocks, until it reaches a full decoder 500 that has the capability of decoding more than one prefix. It is possible, although unlikely, that an instruction has up to fourteen prefix bytes while still being a valid instruction. The prefix decoding circuit includes circuitry to detect the presence of two or more prefixes. In such an instance, a state machine takes over, and sequentially decodes each prefix, one prefix byte at a time, until the entire prefix is decoded. In comparison, the Pentium™ processor treats prefixes as separate instructions, and decodes all in a serial fashion, one per clock, including instances where only one prefix is present.

Each byte from the unsteered instruction buffer 600 is coupled to a prefix decoder 620. Specifically, a first byte is supplied to a first prefix decoder 620a, a second byte is supplied in parallel to a second prefix decoder 620b, and so forth for each byte up to and including a sixteen-bit byte which is supplied in parallel to a sixteen-bit prefix decoder 620p. Within each prefix decoder 620a-p, a byte is decoded as if it were the first and only prefix of an instruction. Specifically, a first prefix decoder 620a supplies a first prefix vector as illustrated at 622a, the second prefix decoder 620b provides a second prefix vector as illustrated at 622b, and so forth, up to and including the sixteen-bit prefix decoder 620p which supplies a sixteen-bit prefix vector 622p. Parallel decoding of all prefix bytes advantageously assures the availability of a prefix vector if the assumption of one prefix in the preferred embodiment turns out to be true. In the preferred embodiment, fifteen rather than sixteen prefix decoders 620a-p are implemented. There is little advantage in decoding the sixteen-bit byte because, if it is a prefix byte, the remainder of the instruction is not available in the 16-byte instruction buffer 600.

Each prefix vector 622a-p includes bits indicating the presence of certain prefix types and is directly usable to indicate the existence of certain prefix bytes. For example, one bit position in the prefix vector may indicate whether the address size is 16- or 32-bits, and another bit position may indicate whether the operand size is 16- or 32-bits. However, some of the bits may not be completely decoded. Which of the bits are fully decoded, and which are partially decoded, is determined by the needs of the circuitry receiving the prefix vector.

Each of the prefix vectors 622a-p, illustrated collectively at 630, are supplied to multiplexers that are coupled to the decoders. Particularly, the prefix vector 630 is coupled to a first 16:1 multiplexer 640 that is coupled to supply a first selected prefix vector illustrated as 642 to the first decoder 120. The prefix vectors 630 are also coupled to a third 16:1 multiplexer 660, which supplies a third selected prefix vector 662 to the third decoder 130.

The prefix logic 610 is coupled to supply control information for the multiplexers 640, 650, and 660. Particularly, in accordance with the byte marks that are being supplied to each of the decoders, and dependent upon the existence of a single prefix byte, a prefix vector will be selected and supplied to the decoder receiving the opcodes associated with that prefix byte. As discussed previously, the selected prefix vectors 642, 652, and 662 are supplied to a field extractor 250 for a partial decoder 200 and a field extractor 570 for a full decoder 500.

FIG. 7 is a flowchart that illustrates operations to decode macroinstructions into Aunops. Beginning in a box 700, a block of instruction code including variable length macroinstructions is loaded into the instruction buffer 600. From the box 700, operation moves in parallel to a box 710, in which macroinstructions are stored to each of the decoders, including, for example the first decoder 110, the second decoder 120, and the third decoder 130. In the box 710, opcodes are stored to the XLAT PLAs in each decoder, and opcodes and following bytes including operands are steered to the field extractor in each decoder. It should be apparent that a separate macroinstruction is steered to each decoder. Steering is performed by steering circuitry 101, to be described in detail with reference to FIGS. 8, 9, 10A, 10B, and 10C.

In parallel with steering as illustrated in the box 710, operation moves to a box 730, the opcodes are supplied to the XLAT PLAs in their respective decoders in which the macroinstructions are translated into a plurality of Cnops. Subsequently, operation moves to a box 734 in which the Cnops are stored in Cnops registers. In the box 740, the opcodes and following bytes including operands from the box 710 and the selected prefix vectors from the box 724 are supplied to field extractors. The selected prefix vectors are associated with opcodes and the following bytes of the corresponding macroinstructions, to extract alias data from each macroinstruction. Subsequently, as illustrated in a box 744, macro-alias data is stored in the macro-alias registers. From the boxes 734 and 744, the Cnops are combined with information in the macro-alias registers and/or macro-alias registers, as illustrated in a box 750, to create Aunops. Subsequently, as illustrated in a box 760, the Aunops from the box 50 are written into the output queue 140.

Instruction Steering

Reference is now made to FIG. 8, which is a block diagram of circuitry for retrieving macroinstructions and

	Docum ent ID	U	Title	Current OR
105	SU 17634 92 A	<input checked="" type="checkbox"/>	Converter tuyeres carriage lifter - with provision for synchronous functioning of both clamps independently giving improved reliability and ease of servicing	
106	JP 04252 631 A	<input checked="" type="checkbox"/>	Synchronisation in switching from one data communication system to other system - loads bus control data for buffer on one system in operation to other system NoAbstract	
107	EP 46855 3 A	<input checked="" type="checkbox"/>	Instrumentation system with down-loading and synchronisation - down loads instructions to instrumentation units and then synchronises in operation by control lines	
108	US 50559 62 A	<input checked="" type="checkbox"/>	Circuit for sync. relay with power to appliance - has micro-controller between power source and relay to ensure contacts are opened or closed at selected point on power line waveform	
109	DE 39424 97 A	<input checked="" type="checkbox"/>	Hydraulic load lifting device - has telescopic hydraulic cylinders designed so that inner and outer parts function simultaneously	
110	JP 01318 595 A	<input checked="" type="checkbox"/>	AC excitation synchronous machine operation - adjusts slip frequency of sync machine by load increase-reduction command power NoAbstract Dwg 1/3	
111	JP 01318 594 A	<input checked="" type="checkbox"/>	AC excitation synchronous machine operation method - supplies load increase-reduction command power to stabilise system frequency NoAbstract Dwg 3/3	
112	GB 22163 06 A	<input checked="" type="checkbox"/>	Multi-processor load and synchronise computer architecture - has semaphore instruction that can be used in parallel processing environment	
113	EP 40626 8 B	<input checked="" type="checkbox"/>	Integrated warehousing system for storing and retrieving goods - has containers delivered from storage carousel brought to temporary queue and held until operator is prepared to work	
114	JP 01145 755 A	<input checked="" type="checkbox"/>	Car audio data transmission - loading voltage sync. instruction pulse on control line, and sending 8 bit data via microcomputer to processor memory NoAbstract DWG 1/7	
115	EP 31038 7 A	<input checked="" type="checkbox"/>	Transmission control apparatus for vehicle - has electric motor actuator controlled by device when shifting gears based on stored stroke value	
116	EP 30975 3 A	<input checked="" type="checkbox"/>	Inductive load monitoring system e.g. for fuel injection valve - evaluates current through series measuring resistance in synch. with operation of load current switch	
117	EP 29789 5 A	<input checked="" type="checkbox"/>	Main memory updating in multiprocessor system - provides read, mask, add, quad word, interlocked instruction to synchronise loading or augmenting of data elements in main memory	
118	US 49655 00 A	<input checked="" type="checkbox"/>	Industrial robot control appts. - has robot operated in shared manner depending on task that is loaded so that automatically works under optimum load condition	
119	EP 26758 3 A	<input checked="" type="checkbox"/>	Turbine helper drive appts. e.g. for steam turbine - has power converter controlling AC motor for generating either positive or negative torque to turbine	
120	GB 21966 03 A	<input checked="" type="checkbox"/>	Movable sheerleg for barge - has pulley at top of sheerleg with cable which extends from winch	
121	US 47315 51 A	<input checked="" type="checkbox"/>	Timed auxiliary power adaptor e.g. for automatic street luminaire - has programmable delay timer synchronised with commencement of main load operation	
122	SU 13700 61 A	<input checked="" type="checkbox"/>	Fork lift truck load remover - has load-item pusher cylinder mounted on truck carriages	
123	JP 63016 028 A	<input checked="" type="checkbox"/>	Gas drier - has towers switched between dehumidification and regeneration by tower switch controller	
124	US 46879 46 A	<input checked="" type="checkbox"/>	Steam generator controller for electric power generator - has digital computer to generate signals to control steam flow during start up, synchronisation and load operation	
125	DD 24617 8 A	<input checked="" type="checkbox"/>	Hydraulic drive for walking conveyor - has several hydraulic motors interconnected by pressure difference valves to allow synchronous operation and no-load operation of individual	
126	SU 12953 79 A	<input checked="" type="checkbox"/>	Pulse type DC voltage stabiliser - has outputs of comparators connected to C-inputs or of first and second D-flip=flops	

[illegible]

	Docum ent ID	U	Title	Current OR
127	SU 12848 60 A	<input checked="" type="checkbox"/>	Self loading vehicle - has pivoting jibs for load handling and additional spring damped levers for control of load oscillation	
128	DE 34476 40 A	<input checked="" type="checkbox"/>	Gear changing method for automatic gearbox - using double clutch operation taking loading on synchronising gearing into account	
129	SU 12200 56 A	<input checked="" type="checkbox"/>	Alternator reactive load distribution and compensation appts. - has self-adjusting system with reactive power sensor in series with memory-comparator and integrating unit	
130	JP 61049 681 A	<input checked="" type="checkbox"/>	Current instruction generator for sync. motor - corrects load angle instruction to achieve high-precision load angle control and high response torque control NoAbstract Dwg 1/4	
131	SU 12118 44 A	<input checked="" type="checkbox"/>	Control system for stepper motor - with phases divided into two groups, control pulse being applied to two phases, only one being connected	
132	JP 61010 912 A	<input checked="" type="checkbox"/>	Protective-relay for synchronous power generator - determines plane-coordinates of active and reactive powers in load operation by judging in-out sides NoAbstract Dwg 11/16	
133	SU 12030 10 A	<input checked="" type="checkbox"/>	Crane load suspension with crab and pulley ropes - has additional large dia. pulleys coaxial with main crossbar pulleys and specified rope path for rope break protection	
134	US 45368 49 A	<input checked="" type="checkbox"/>	Machine operation monitor to prevent tool breakage - responds to synchronisation signals to maintain correspondence between load level samples from successive cycles	
135	US 45361 26 A	<input checked="" type="checkbox"/>	Synchronisation between gas turbine and power system - has computer program and external logic circuitry operating in control loop	
136	DE 34203 70 A	<input checked="" type="checkbox"/>	Single phase induction motor with AC electromagnetic stator - has inner freely rotting permanent magnet synchronous rotor and outer load connected asynchronous cage rotor	
137	SU 11569 88 A	<input checked="" type="checkbox"/>	Loading device for conveyor load carrier - has fixed cam acting with roller on scoop reloader which is mounted on running screw of drive mechanism	
138	SU 11557 48 A	<input checked="" type="checkbox"/>	Chainless haulage for face machines - has rotating housings of drive sprockets connected together by pin and bush	
139	US 44880 58 A	<input checked="" type="checkbox"/>	Snubber circuit for uninterruptable power supply - has MOSFET whose gate electrode is connected to transformer receiving control signals to dissipate current transient	
140	EP 12896 1 A	<input checked="" type="checkbox"/>	Synchronous motor having pole outer surface length - determined w.r.t. slots per pole per phase ratio and slot pitch	
141	DE 32469 30 A	<input checked="" type="checkbox"/>	Load switching between mains and back-up supply - has synchronised operation of switches for mains and back-up supply	
142	GB 21304 49 A	<input checked="" type="checkbox"/>	Time switch for electricity meter - has synchronous operation with two tariff supply to control load, e.g. water heater, and use off-peak power	
143	EP 10385 1 A	<input checked="" type="checkbox"/>	Multiprocessor control synchronisation and instruction down-loading - provides input-output processor with immediate access to timing and sync. signals for prompt response to control of master processor	
144	DE 33036 07 A	<input checked="" type="checkbox"/>	Synchronisation control for load raising elements - uses microprocessor monitoring effected lift and controlling operation of individual raising elements	
145	US 43801 46 A	<input checked="" type="checkbox"/>	Gas turbine electric power plant control system - uses programmed digital computer control system during sequenced start-up, synchronising, load and shut-down operations	
146	EP 73280 A	<input checked="" type="checkbox"/>	Gear shift for vehicle - has main change speed gear with simultaneously actuated hydraulic clutch	
147	SU 98498 7 A	<input checked="" type="checkbox"/>	Wheeled loader - has synchronised bucket and chute with adjustment for increased output	
148	DE 32117 97 C	<input checked="" type="checkbox"/>	Intermittent driving device for e.g. manufacturing machines - has computer to control speed of second load of two-load system according to speed and position of first load	

instruction block within the instruction buffer 840 is now aligned with the first macroinstruction.

Reference is now made to FIGS. 10A and 10B, and 10C which together are a flow chart illustrating the procedure for steering instructions to the decoder.

Beginning with a start block 1000, the procedure for steering instructions to the decoder.

steering instructions begins. Operation moves to a block 1010, in which the first opcode byte marks are scanned to locate the first opcode byte of a first instruction. In parallel, operation moves to block 1012 in which the first opcode byte

marks are scanned to locate the first opcode byte of a second instruction. Also in parallel, operation moves to a block 1014 in which the first opcode byte marks are scanned to

locate the first opcode byte of a third instruction. Beginning at the first opcode byte from the block 1010, operation moves to block 1020 in which 11 bytes of instruction code

are steered to the first decoder, beginning at the located first opcode byte. From the box 1012, 8 bytes of instruction code

are steered to the second decoder, the 8 bytes beginning at the first opcode byte of the second instruction. Similarly, for

the third decoder 130, as illustrated in an operation 1024, the third decoder 130, the associated prefix vector is supplied to

in a block 1032, the associated prefix vector is supplied to the first decoder 110. Similarly, from the block 1012, operation moves to the box 1034 in which a prefix vector is

associated with the second instruction, and in a box 1036 the prefix vector is supplied to the second decoder 120. Simi-

larly, for the third decoder, from the box 1014 and a prefix vector is associated with a box 1038 with the third instruction, and then a box 1039 the prefix vector is supplied to the

third decoder 130.

One feature of the steering mechanism described herein is the ability to resume instructions, dependent upon what happens in subsequent operations. When the first instruction

has been steered from the boxes 1020 and 1032 to the first decoder 110, then the decision defined in the decision box

1050 determines whether or not that first instruction is decodable in one decode cycle. Macroinstructions for which the first instruction are not decodable in one decode cycle

include, for example, an instruction that requires an entry into microcode, which requires at least two clock cycles in which to perform its operations. Another example is an illegal opcode, which requires at least two cycles in which to respond. Therefore, if the first instruction is not decodable

in one cycle, then operation moves from the decision 1050 to a block 1052 which indicates that instruction decoding in the second and third decoders is stopped and the instructions

therein flushed to prevent micro-operations being issued therefrom. Subsequently, operations are performed to decode the first instruction in multiple decode cycles. The

decoding operation proceeds as described in detail previously.

However, if from the box 1050, the first instruction is decodable in one cycle, then the second decoder 120 and the third decoder 130 can provide useful micro-operations. In that instance, operation moves from the decision 1050 in parallel to a decision 1060 which queries whether the second instruction is decodable by the second decoder 120, and a decision 1062 which queries whether the third instruction is

decodable by the third decoder 130. Furthermore, in parallel, the first instruction is decoded as illustrated in a box 1064. From the decision 1060, if the second instruction is not

decodable by the third decoder 130, the first instruction is not decodable in one cycle, then the second decoder 120 and the third decoder 130 can provide useful micro-operations. In that instance, operation moves from the decision 1050 in parallel to a decision 1060 which queries whether the second instruction is decodable by the second decoder 120, and a decision 1062 which queries whether the third instruction is

decodable by the third decoder 130. Furthermore, in parallel, the first instruction is decoded as illustrated in a box 1064. From the decision 1060, if the second instruction is not

decodable by the third decoder 130, the first instruction is not decodable in one cycle, then the second decoder 120 and the third decoder 130 can provide useful micro-operations. In that instance, operation moves from the decision 1050 in parallel to a decision 1060 which queries whether the second instruction is decodable by the second decoder 120, and a decision 1062 which queries whether the third instruction is

decodable by the third decoder 130. Furthermore, in parallel, the first instruction is decoded as illustrated in a box 1064. From the decision 1060, if the second instruction is not

decodable by the third decoder 130, the first instruction is not decodable in one cycle, then the second decoder 120 and the third decoder 130 can provide useful micro-operations. In that instance, operation moves from the decision 1050 in parallel to a decision 1060 which queries whether the second instruction is decodable by the second decoder 120, and a decision 1062 which queries whether the third instruction is

decodable by the third decoder 130. Furthermore, in parallel, the first instruction is decoded as illustrated in a box 1064. From the decision 1060, if the second instruction is not

decodable by the third decoder 130, the first instruction is not decodable in one cycle, then the second decoder 120 and the third decoder 130 can provide useful micro-operations. In that instance, operation moves from the decision 1050 in parallel to a decision 1060 which queries whether the second instruction is decodable by the second decoder 120, and a decision 1062 which queries whether the third instruction is

decodable by the third decoder 130. Furthermore, in parallel, the first instruction is decoded as illustrated in a box 1064. From the decision 1060, if the second instruction is not

decodable by the third decoder 130, the first instruction is not decodable in one cycle, then the second decoder 120 and the third decoder 130 can provide useful micro-operations. In that instance, operation moves from the decision 1050 in parallel to a decision 1060 which queries whether the second instruction is decodable by the second decoder 120, and a decision 1062 which queries whether the third instruction is

decodable, then operation moves to the operation box 1070. The operation box 1070 also is coupled to the box 1060. Therefore, to get to the box 1070, the first instruction must be decoded, and the second instruction is not decodable by the second decoder 120. Subsequently, operation moves to box 1000 which is the start of the procedure. The operations herein are repeated to steer the former second instruction to the first decoder 110, the former third instruction is steered to the second decoder 120, and any subsequent undecoded instruction in the instruction buffer 840 is steered to the third decoder 130.

However, if from the decision box 1060, it is determined that the second instruction is decodable by the second decoder 120, then operation moves to a block 1080 which illustrates that the second instruction is decoded. From the box 1080, a usable micro-operation will be provided. From the decision 1062, if the third instruction is decodable by the third decoder 130, then operation moves to the box 1082 which illustrates that the third instruction is decoded. Therefore, from the boxes 1064, 1080, and 1082, operation moves to box 1086, which shows that the first, second and third instructions have been decoded.

However, if from the decision box 1062, the third instruction is not decodable by the third decoder, then operation moves to the box 1088, which illustrates that the first and second instructions have been decoded, but the third instruction is not decodable by the third decoder 130. Subsequently, operation moves to a box 1090, which shows that operations then return to the start of the procedure, to steer the third instruction to the first decoder 110. Any subsequent undecoded instructions in the instruction buffer 840 are then steered to the second decoder 120 and the third decoder 130. If all instructions have been decoded (each have provided valuable Cnops), as illustrated in a box 1086, then operation moves to a decision 1092, which is useful for determining when additional instructions must be supplied to the instruction buffer 840. Particularly, the decision box 1092 asks whether all instructions in the instruction buffer 840 have been decoded. If not, then operation moves to a box 1094 in which operation returns to the start, and the steering operation is repeated for all undecoded instructions. However, if from the box 1092 all instructions have been decoded, then operation moves to the box 1096 which indicates that the steering procedure is stopped and is now complete. In other words, all complete instructions within the instruction buffer 840 have been consumed by the decoder and the next 16-byte block of instruction code can be loaded into the instruction buffer 840. Reference is again briefly made to FIG. 9, which shows that another 16-byte block of instruction code is loaded into the instruction buffer 840 when the steering procedure is complete.

Assist Handling

Reference is now made to FIG. 11 which is a flow chart illustrating assist handling including state recovery. The flow chart of FIG. 11 also illustrates decoder state recovery associated with macroinstruction restart. Beginning in a start block 1100, operation starts when an assist routine is called in microcode as a result of an error that has been recognized after execution of a micro-operation. This includes errors such as faults or exceptions. Next, as illustrated in blocks 1102, the assist routine begins executing. Execution includes issuing assist handling micro-operations from the MS unit. Operation then moves to the box 1105. For some assist routines, recovery of decoder state will be necessary. For example, if the assist routine requires data from an error-causing macroinstruction, then state recovery begins by

	Docum ent ID	U	Title	Current OR
149	EP 25087 A	<input checked="" type="checkbox"/>	Pipelined micro-instruction generator for digital processor - has pipeline register responsive to branch instruction to prevent loading for execution of aborted sequential instruction	
150	SU 79392 4 B	<input checked="" type="checkbox"/>	Two synchronised cylinders for load lifting device - has hydraulically controlled two-position, two-line distributor, with hydraulic lines connected to cylinders chambers	
151	SU 77925 2 B	<input checked="" type="checkbox"/>	Load-lifting grab for traversing crane - has operations synchroniser as two rollers and two intersecting rods with their ends attached to rollers	
152	SU 77535 2 B	<input checked="" type="checkbox"/>	Steam turbine regulator system with steam bleeding - has synchroniser slide valve linked in series to speed regulator slide valve for faster operation	
153	SU 76599 6 B	<input checked="" type="checkbox"/>	Observation device with pulse phase regulator - has phase comparator to synchronise synchronisation generator with mains frequency	
154	SU 75412 8 B	<input checked="" type="checkbox"/>	Two hydraulic cylinder rods synchroniser - has distributors in pressure lines which connect cylinder piston spaces to drains and include additional manually closed distributors	
155	DE 29047 86 B	<input checked="" type="checkbox"/>	Voltage regulation system with load distribution in parallel inverters - uses derived control parameter by comparison with reference and impedance	
156	DE 29459 67 A	<input checked="" type="checkbox"/>	Synchromesh gearbox for road vehicle - has supporting elements inside holes for synchronising cone springs to prevent them distorting	
157	DE 29459 66 A	<input checked="" type="checkbox"/>	Synchromesh motor vehicle gearbox - has holes for synchronising cone springs passing through deeper part of hub to provide greater support	
158	SU 73152 3 B	<input checked="" type="checkbox"/>	Transformer load antiparallel thyristors control appts. - has current sensor in magnetic memory to operate trigger so that thyristor is turned on by synchronising signal	
159	US 41744 96 A	<input checked="" type="checkbox"/>	Monolithic solid state power controller - detects zero voltage crossing in source voltage and pre-selects mode synchronisation	
160	SU 65757 2 A	<input checked="" type="checkbox"/>	Single-phase thyristor inverter control - by returning reactive load current with advance w.r.t. high voltage supply by thyristor reset angle	
161	SU 61999 4 A	<input checked="" type="checkbox"/>	Ship shaft generators parallel operation switching - by distributing electrical power to mechanical load, actuating decoupling clutch and equalising fuel and electrical power supplies	
162	NL 77051 99 A	<input checked="" type="checkbox"/>	Lift truck independent transmissions - is for load carrier rotation and translation coupled to permit phase synchronised operation	
163	US 40314 07 A	<input checked="" type="checkbox"/>	Electric power plant operating system - has hybrid subsystem including computer and external phase detection circuitry for automatic synchronization	
164	US 40193 15 A	<input checked="" type="checkbox"/>	Gas turbine power plant control - includes temp. reset starting and ignition pressure control systems with digital function capability	
165	SU 54161 6 A	<input checked="" type="checkbox"/>	Single phase contact welder - with automatic phasing of current switching by thyristors for stabilisation of loading	
166	US 39991 69 A	<input checked="" type="checkbox"/>	Real time control for digital computer - uses real time clock placed in central processor and modifiable with memory reference instructions	
167	SU 51856 6 A	<input checked="" type="checkbox"/>	Mine heading machine tooling - has vee rollers on shafts linked through gear and centre shaft to main cutter drum drive	
168	US 39596 35 A	<input checked="" type="checkbox"/>	System for operating steam turbine - has digital computer control with improved automatic startup control features	
169	DE 25435 86 A	<input checked="" type="checkbox"/>	Optional operation of electrical loads - selector circuit produces pulse of certain length synchronously with hand switch operation	
170	SU 49065 0 A	<input checked="" type="checkbox"/>	Circular loaf slicer - with rotary table linked by shaft and lever to overrun clutch	

restoring macro-alias data to macro-alias registers and the first Cuops to the Cuop registers. This operation, termed "restore MAR", is discussed with reference to FIG. 12. Briefly, the effect of the restore MAR operation is to reload the macro-alias registers with fields extracted from the assisted macroinstruction so that it can be utilized by subsequent code in the assist handling routine or other code, such as restarted code, as will be described. For this purpose, the restore MAR operation is inserted into the assist handling routine at a location in which can be properly utilized. For example, if the assist handling routine does not require use of the macro-alias data until subsequent restart, then the restore MAR in the box 1105 can be delayed until the end of the assist handling routine. However, if the macro-alias data is needed by the assist handling routine, then the operations to restore MAR operation in the box 1105 will occur before such instructions, and possibly as early as the beginning of the assist handling routine (or not done at all, if not required by the assist).

Additional state recovery may also be necessary to restore the state of the macro-alias registers. For example, operations to recover decoder state may be required if a macroinstruction subsequent to the restart target uses macro-alias data from previous micro-operations. If necessary for use by subsequent code, the restore UAR routine, illustrated in FIG. 14 and discussed with reference thereto, is called to restore the contents of macro-alias data for use by subsequent micro-operations.

After the macro-alias data and the micro-alias data has been restored, as necessary, in the box 1105, then control is transferred back to the MS unit as illustrated in a box 1106, to continue the assist handling routine and, in some instances, to restart, as will be described.

After the assist handling sequence has performed its operations, it may be advantageous to restart the assisted macroinstruction. Restart is then directed by the last step of the assist handling routine, which will then transfer control to the microcode beginning at the restart target.

The restart decision is illustrated in a box 1110. If no further restart is desired at the end of the assist handling sequence, then operation moves to the box 1112 which indicates that the assist handling routine is exited and therefore operations continue wherever indicated by the assist handling sequence.

If restart is desired from the decision box 1110, then operation moves to the box 1120 which illustrates that the decoder state has been recovered in the box 1105. However, if additional state recovery were to be required, then the appropriate operations, discussed above, could be performed. The restart target can be at any micro-operation within the micro-operation flow, which may be one of the first several Cuops issued from the XLAT PLAs, or it may be the position of the first Cuop that will eventually be issued from the decoder. The decoder then asserts the valid bit for the target Cuop and subsequent Cuops in the Cuop register, which then allows these Cuops to be issued to restart the assisted macroinstruction.

If necessary, the MS unit 534 will supply subsequent micro-operations to continue the micro-operation flow. This operation moves to the box 1310, in which microcode is

method is particularly advantageous if, for example, the third and fourth Cuops will be executed during a restart, but the first and second Cuops will not. Therefore, the described method is useful for restarting at any Cuop in the flow from a macroinstruction.

If from the decision 1130, restart begins at one of the Cuops within the MS unit 534, then operation moves from the decision 1130 to the block 1140 in which Cuops are supplied from the MS unit 534 to restart the assisted macroinstruction.

Reference is now made to FIG. 12, which illustrates operations to restore the macro-alias data. The operation illustrated in FIG. 12 is termed "restore MAR". Operation begins in a box 1200 which indicates the start of the restore MAR operation to restore macro-alias data in the macro-alias registers, including restoring the first Cuops that can be generated in the XLAT PLAs.

Following start, operation moves to a box 1210 in which a signal is supplied from the MS unit 534 to fetch the designated macroinstruction. This signal may include a LOADMAR ("Load Macro-Alias Register") signal. In the event an assist is in process, the designated macroinstruction will be the macroinstruction that originated the micro-operation sequence for which the assist is required. If the restore MAR operation is occurring as a result of a mispredicted micro-branch, then the designated macroinstruction is the macroinstruction that supplied the micro-operation sequence in which the mispredicted branch occurred.

From the box 1210, operation moves to the box 1220 in which the designated macroinstruction is supplied to the decoder 500. In the preferred implementation, the decoder in which this operation can be performed is the full decoder 500, because it is the only decoder that is coupled to a microcode sequencing unit.

From the box 1220, operation moves in parallel to the box 1230 in which the opcodes from designated macroinstructions are decoded in the XLAT PLAs and then, as illustrated in a box 1232, the Cuops resulting therefrom are supplied to the Cuops register with their valid bit unasserted. If a valid bit is unasserted (i.e., marked "invalid"), the Cuops will not be issued into subsequent units.

From the box 1220, operation also moves to a box 1240 in which fields are extracted within the field extractor 570, and then, as illustrated in a box 1242 macro-alias data 578 is supplied to macro-alias registers.

After the above operations have been completed, the restore MAR operation is complete as illustrated in a box 1250. As a result of the restore MAR operations, the macro-alias registers are restored, and the Cuops are also restored so that the initial Cuops from the XLAT PLAs are stored therein, but with their valid bits remaining unasserted (i.e., marked invalid).

Reference is now made to FIG. 13, which is a flow chart that illustrates operations to restore the micro-alias register. This operation is termed "restore UAR". The restore UAR operation begins in a box 300, which indicates that the micro-alias data is to be restored in a micro-alias register. This routine is particularly useful for writing generic assist or fix-up microcode routines. For example, a routine that fixes up an underflow or overflow error can load the micro-alias registers from the problematic Cuop and reference its sources and destination fields to repair the result. The restore UAR operation is useful to restore micro-alias data from any micro-operation that can then be referenced in generic routines that save microcode space.

From the operation 1300 to start the restore UAR routine, operation moves to the box 1310, in which microcode is

	Docum ent ID	U	Title	Current OR
171	US 39241 41 A	<input checked="" type="checkbox"/>	Gas turbine power plant control apparatus - system has monitoring and protective sub systems functioning through all operational stages	
172	DE 21040 76 B	<input checked="" type="checkbox"/>	Step transformer thyristor load switching circuit - has logic to synchronise antiparallel thyristor ignition to series and parallel circuit breakers operations	
173	DE 21040 75 B	<input type="checkbox"/>	Step transformers thyristor load switching circuit - has logic to synchronise antiparallel thyristor ignition to series and parallel circuit breaker operations	

retrieved that generated the micro-operation having the data to be stored in the micro-alias register. Subsequently, from the box 1320, the micro-operation is generated by the decoder but not issued therefrom. The micro-operation can be generated by microcode or it can be generated in the XIAT PLAs. In the latter case, the micro-operations are generated by the restore MIAK operation discussed above. Then, operation moves to the box 1330 in which the micro-alias data from that micro-operation is saved into the micro-alias register. Then, the restore UAK operation is complete, and the micro-alias data is restored for use by subsequent micro-operations.

Reference is now made to FIG. 14 which illustrates operations to restart a micro-operation flow at an actual micro-branch target. The operations illustrated herein typically result from a mispredicted micro-branch. As defined herein, a micro-branch is a branch made by microcode. Operation begins in a box 1400 in which an actual micro-branch target is known and restart is desired at that actual target. Operation then moves to the box 1410 which indicates that the control goes to the target address at the beginning of the target microcode flow.

Subsequent operations illustrated in boxes 1420 and 1430 restore the micro-alias registers, as described with reference to FIG. 12. If either micro-alias data or micro-alias data or both may be used by subsequent micro-operations, then the first instructions at the target microcode flow restore the state by restoring the micro-alias registers if necessary and by restoring the micro-alias registers if necessary for use by subsequent micro-operations.

From the box 1430, operation then moves to execute the subsequent microcode flow.

What is claimed is:

1. A method for assembling a micro-operation in a decoder, comprising the steps of:
 - a. decoding a macroinstruction into a plurality of intermediate micro-operations including a source immediate micro-operation and a subsequent immediate micro-operation;
 - b. storing a selected field of the source immediate micro-operation in a micro-alias register, wherein the selected field of the source immediate micro-operation is independent of any operand of the macroinstruction;
 - c. assembling an alias micro-operation from the subsequent immediate micro-operation and the selected field stored in the micro-alias register.
2. A method of decoding a macroinstruction, comprising the steps of:
 - a. decoding the macroinstruction into a plurality of intermediate micro-operations including a source immediate micro-operation and a subsequent immediate micro-operation;
 - b. storing a selected field of the macroinstruction in a macro-alias register as macro-alias data;
 - c. storing a selected field of the source immediate micro-operation in a micro-alias register as micro-alias data, wherein the micro-alias data is independent of any operand of the macroinstruction;
 - d. assembling an alias micro-operation from the subsequent immediate micro-operation and both the micro-alias data and the macro-alias data.

3. A method of decoding a macroinstruction, comprising the steps of:

- a) decoding at least one opcode of the macroinstruction into a plurality of control micro-operations in a control micro-operation flow, the plurality of control micro-operations and a subsequent control micro-operation operations including a source control micro-operation and a subsequent control micro-operation;
- b) selecting at least one field from the source control micro-operation;
- c) storing the at least one field from the source control micro-operation in a micro-alias storage device as micro-alias data, wherein the micro-alias data is independent of any operand of the macroinstruction;
- d) assembling an alias micro-operation by combining the subsequent control micro-operation with the micro-alias data.

4. The method of claim 3 wherein step a) comprises:

- i) supplying the opcode to a translate programmable logic array to provide the source control micro-operation;
- ii) supplying the opcode to an entry point logic device so that the entry point logic device supplies an entry point address signal; and
- iii) supplying the entry point address signal to a microcode sequencing unit to generate a plurality of microcode control micro-operations in the control micro-operation flow, the microcode control micro-operations including the source control micro-operation and the subsequent control micro-operation.

5. The method of claim 3 wherein step a) comprises:

- i) supplying the opcode to an entry point logic device so that the entry point logic device supplies an entry point address signal; and
- ii) supplying the entry point address signal to a microcode sequencing unit to generate a plurality of microcode control micro-operations in the control micro-operation flow, the microcode control micro-operations including the source control micro-operation and the subsequent control micro-operation.

6. A computer processing method for storing information from a first control micro-operation (Cuop) of a Cuop flow for use by a subsequent second Cuop in the Cuop flow, said method comprising the steps of:

- a) supplying the first Cuop from a translate programmable logic array (XLAT PLA) in response to a macroinstruction supplied to the XLAT PLA;
- b) applying the macroinstruction in parallel with the step a) to a field extractor which extracts macro-data fields therefrom;
- c) storing micro-alias data corresponding to predetermined fields from the first Cuop in a micro-alias register, wherein the micro-alias data is independent of any operand of the macroinstruction;
- d) supplying the first Cuop and said macro-alias data to an alias multiplexer to resolve aliases in the first Cuop to assemble a first output micro-operation (Auop); and
- e) supplying the second Cuop and the micro-alias data to the alias multiplexer to resolve aliases in the second Cuop to assemble a second Auop.

7. The computer processing method of claim 6, wherein, if a restart is required at an actual micro-target address, then performing the following steps:

- a) transferring control to the micro-target address at the beginning of a micro-target flow, the micro-target microcode flow having a first microcode Cuop and subsequent microcode Cuops, wherein at least one

	Docum ent ID	U	Title	Current OR
1	JP 20031 09381 A	<input type="checkbox"/>	BOOSTING POWER SOURCE GENERATING CIRCUIT	
2	JP 20031 02194 A	<input checked="" type="checkbox"/>	DRIVE DEVICE FOR BRUSHLESS MOTOR	
3	JP 20030 87995 A	<input checked="" type="checkbox"/>	AUXILIARY POWER SUPPLY DEVICE	
4	JP 20030 70293 A	<input checked="" type="checkbox"/>	CONTROLLER OF PERMANENT MAGNET SYNCHRONOUS MACHINE	
5	JP 20030 61347 A	<input checked="" type="checkbox"/>	SWITCHING POWER SUPPLY	
6	JP 20022 84363 A	<input checked="" type="checkbox"/>	PAPER FEEDING DEVICE AND IMAGE FORMING DEVICE	
7	JP 20022 23434 A	<input checked="" type="checkbox"/>	VIDEO MONITORING SYSTEM AND VIDEO MONITORING METHOD	
8	JP 20022 09344 A	<input checked="" type="checkbox"/>	NONCONTACT POWER TRANSMISSION APPARATUS	
9	JP 20021 56037 A	<input checked="" type="checkbox"/>	TRANSMISSION MODE SWITCHING CONTROL DEVICE OF TRANSMISSION WITH INFINITE CHANGE GEAR RATIO	
10	JP 20020 13427 A	<input checked="" type="checkbox"/>	SPEED CONTROLLER OF ENGINE FOR GENERATOR	
11	JP 20013 39997 A	<input checked="" type="checkbox"/>	AUTOMATIC REGULATOR OF SYNCHRONOUS GENERATOR	
12	JP 20013 27187 A	<input checked="" type="checkbox"/>	METHOD FOR CONTROLLING BRUSHLESS MOTOR	
13	JP 20013 00794 A	<input checked="" type="checkbox"/>	SYNCHRONOUS DRIVE CONTROL METHOD OF PRESS, AND PRESS USED THEREFOR	
14	JP 20013 00793 A	<input checked="" type="checkbox"/>	SYNCHRONOUS DRIVE CONTROL METHOD FOR PRESS, AND THE PRESS USED THEREFOR	
15	JP 20013 00792 A	<input checked="" type="checkbox"/>	SYNCHRONOUS DRIVE CONTROL METHOD FOR PRESS, AND THE PRESS USED THEREFOR	
16	JP 20012 32853 A	<input checked="" type="checkbox"/>	IMAGE FORMING APPARATUS, METHOD OF CONTROLLING SIGNAL AND MEMORY MEDIUM	
17	JP 20012 04192 A	<input checked="" type="checkbox"/>	CONTROL UNIT OF BRUSHLESS MOTOR AND SELF-PRIMING PUMP USING THE SAME	

	Docum ent ID	U	Title	Current OR
18	JP 20011 96193 A	<input checked="" type="checkbox"/>	POWER EQUIPMENT, DISCHARGE LAMP LIGHTING EQUIPMENT AND LIGHTING SYSTEM	
19	JP 20011 78184 A	<input checked="" type="checkbox"/>	INVERTER DEICE AND ELECTRIC WASHING MACHINE INCORPORATING THE SAME	
20	JP 20010 89094 A	<input checked="" type="checkbox"/>	FORKLIFT	
21	JP 20010 03864 A	<input checked="" type="checkbox"/>	AIR CONDITIONER	
22	JP 20003 54399 A	<input checked="" type="checkbox"/>	SECONDARY EXCITATION CONTROL METHOD FOR AC EXCITED SYNCHRONOUS MACHINE	
23	JP 20002 67850 A	<input checked="" type="checkbox"/>	CONTROLLER DEALING WITH INTERACTIVE	
24	JP 20001 65228 A	<input checked="" type="checkbox"/>	LOGIC CIRCUIT	
25	JP 20001 30463 A	<input checked="" type="checkbox"/>	SYNCHRONIZING DEVICE FOR GEAR TRANSMISSION	
26	JP 11316 019 A	<input checked="" type="checkbox"/>	ENERGIZING CIRCUIT FOR COMBUSTOR	
27	JP 11314 675 A	<input checked="" type="checkbox"/>	CONTAINER FOR VACUUM PACKAGING, AND VACUUM PACKAGING METHOD USING IT	
28	JP 11311 583 A	<input checked="" type="checkbox"/>	VIBRATING DEVICE	
29	JP 11211 078 A	<input checked="" type="checkbox"/>	CURRENT APPLICATION CIRCUIT OF COMBUSTOR	
30	JP 11167 557 A	<input checked="" type="checkbox"/>	SHARED MEMORY ACCESS SEQUENCE ASSURANCE METHOD AND MULTIPROCESSOR SYSTEM	
31	JP 11103 599 A	<input checked="" type="checkbox"/>	STARTER FOR GENERATOR	
32	JP 11002 441 A	<input checked="" type="checkbox"/>	METHOD AND DEVICE FOR OPERATION OF FLOOR HEATING SYSTEM	
33	JP 10326 274 A	<input checked="" type="checkbox"/>	DIGITAL CONTENT EDITION METHOD AND DEVICE THEREFOR AND RECORDING MEDIUM FOR RECORDING DIGITAL CONTENT EDITION PROGRAM	
34	JP 10299 893 A	<input checked="" type="checkbox"/>	TRANSMISSION-OPERATING BOOSTER	
35	JP 10286 000 A	<input checked="" type="checkbox"/>	CONTROLLER FOR SYNCHRONOUS MOTOR	
36	JP 10230 659 A	<input checked="" type="checkbox"/>	IMAGE RECORDER	
37	JP 10226 106 A	<input checked="" type="checkbox"/>	APPARATUS AND METHOD FOR FORMING IMAGE	

data and the micro-alias data to resolve aliases in the selected Cuop to provide an output micro-operation (Auop).

14. The micro-operation aliasing mechanism of claim 13 wherein, for at least one macroinstruction that is decoded into a plurality of Cuops including a first Cuop and subsequent Cuops, the XLAT supplies the first Cuop from the macroinstruction and the microcode ROM supplies the subsequent Cuops so that, in a first cycle, the first Cuop is supplied from the XLAT PLA, and in a second cycle, the subsequent Cuops are supplied from the MSU.

15. The micro-operation aliasing mechanism of claim 14 further comprising:

a micro-alias register storing means responsive to at least one of the subsequent Cuops for loading the micro-alias register with the micro-alias data from the first Cuop. 16. A macroinstruction decoder for decoding a macroinstruction having at least one opcode, comprising: a translate programmable logic array for supplying at least one control micro-operation in response to the macroinstruction, if the opcode belongs to a first predefined set of opcodes; a field extraction circuit coupled to receive the macroinstruction in parallel with the translate programmable logic array, the field extraction circuit extracting macro-alias data from the macroinstruction;

a macro-alias register for storing the macro-alias data; a microcode sequencing unit for supplying at least one control micro-operation in response to the macroinstruction, if the opcode belongs to a second predefined set of opcodes;

a micro-alias register selecting a selected control micro-operation from one of the translate programmable logic array and the microcode sequencing unit in response to a control signal from the microcode sequencing unit, wherein a micro-alias data from the selected control micro-operation is loaded into the micro-alias register in response to a load signal from the microcode sequencing unit, wherein the micro-alias data is independent of any operand of the macroinstruction; and an alias multiplexer, wherein the alias multiplexer combines a subsequent control micro-operation with at least one of the micro-alias data from the micro-alias register and the macro-alias data from the macro-alias register to form an aliased micro-operation in accordance with the subsequent control micro-operation, the subsequent control micro-operation provided from one of the translate programmable logic array and the microcode sequencing unit.

* * * * *

	Docum ent ID	U	Title	Current OR
38	JP 10214 073 A	<input checked="" type="checkbox"/>	DISPLAY DEVICE	
39	JP 10149 888 A	<input checked="" type="checkbox"/>	LIGHTING DEVICE FOR ILLUMINATION AND ITS CONTROL METHOD	
40	JP 10136 682 A	<input checked="" type="checkbox"/>	CONSTANT TORQUE SYNCHRONOUS DRIVE CONTROL EQUIPMENT	
41	JP 10127 088 A	<input checked="" type="checkbox"/>	STARTER AND STARTING METHOD FOR DC BRUSHLESS MOTOR	
42	JP 10106 786 A	<input checked="" type="checkbox"/>	POWER SUPPLY DEVICE	
43	JP 10094 290 A	<input checked="" type="checkbox"/>	LOAD CONTROL APPARATUS	
44	JP 10054 753 A	<input checked="" type="checkbox"/>	CARRYING LOAD RECORDING DEVICE	
45	JP 10032 350 A	<input checked="" type="checkbox"/>	DRIVING CIRCUIT	
46	JP 09331 680 A	<input checked="" type="checkbox"/>	LOAD DRIVING CIRCUIT	
47	JP 09261 628 A	<input checked="" type="checkbox"/>	IMAGE COMPRESSOR, IMAGE EXPANDER, AND IMAGE COMPANDER	
48	JP 09217 828 A	<input checked="" type="checkbox"/>	BOOSTING DEVICE FOR OPERATING TRANSMISSION	
49	JP 09215 371 A	<input checked="" type="checkbox"/>	SYNCHRONOUS CONTROLLER	
50	JP 09214 464 A	<input checked="" type="checkbox"/>	SYNCHRONISM DETECTION DEVICE FOR ORTHOGONAL FREQUENCY DIVISION MULTIPLEX RECEIVER	
51	JP 09201 098 A	<input checked="" type="checkbox"/>	AUTOMATIC VOLTAGE REGULATION APPARATUS OF PARALLEL COMPENSATION TYPE	
52	JP 09138 209 A	<input checked="" type="checkbox"/>	METHOD AND APPARATUS FOR DETECTION OF GAS	
53	JP 09120 388 A	<input checked="" type="checkbox"/>	METHOD AND SYSTEM FOR INFORMATION PROCESSING	
54	JP 09077 272 A	<input checked="" type="checkbox"/>	SHEET MATERIAL CONVEYING DEVICE	
55	JP 09068 737 A	<input checked="" type="checkbox"/>	BARRIER OPENING AND CLOSING MECHANISM	
56	JP 09019 109 A	<input checked="" type="checkbox"/>	OVERHEAT PROTECTOR FOR FIELD COIL	
57	JP 08322 285 A	<input checked="" type="checkbox"/>	CONTROL METHOD OF BRUSHLESS MOTOR	
58	JP 08289 539 A	<input checked="" type="checkbox"/>	SWITCHING POWER SUPPLY SYSTEM	
59	JP 08147 016 A	<input checked="" type="checkbox"/>	SERVO DEVICE FOR MULTI-AXIS OPERATION	
60	JP 08132 336 A	<input checked="" type="checkbox"/>	PRODUCTION INSTRUCTING DEVICE	

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 5,559,974

DATED : September 24, 1996

INVENTOR(S) : Boggs et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 16 at line 64 delete "50" and insert --750--

Signed and Sealed this

Eleventh Day of March, 1997

Bruce Lehman

BRUCE LEHMAN

Commissioner of Patents and Trademarks

Attesting Officer

	Docum ent ID	U	Title	Current OR
61	JP 08121 315 A	<input checked="" type="checkbox"/>	PELTON WHEEL	
62	JP 08009 693 A	<input checked="" type="checkbox"/>	SECONDARY EXCITATION CONTROL METHOD OF AC EXCITATION SYNCHRONOUS MACHINE	
63	JP 07301 136 A	<input checked="" type="checkbox"/>	METHOD AND ARRANGEMENT FOR CONTROLLING INTERNAL COMBUSTION ENGINE	
64	JP 07248 847 A	<input checked="" type="checkbox"/>	METHOD AND DEVICE FOR ADJUSTING CLOCK SIGNAL	
65	JP 07164 358 A	<input checked="" type="checkbox"/>	ROBOT CONTROL DEVICE	
66	JP 07107 542 A	<input checked="" type="checkbox"/>	SYSTEM FOR DOWN LOAD DURING SYNCHRONOUS OPERATION	
67	JP 07053 100 A	<input checked="" type="checkbox"/>	WINDING TENSILE FORCE CONTROL DEVICE	
68	JP 07030 619 A	<input checked="" type="checkbox"/>	SYNCHRONIZING CONTROL METHOD FOR SEMI-DUAL COMMUNICATION TEST	
69	JP 07028 602 A	<input checked="" type="checkbox"/>	DIGITAL ARITHMETIC UNIT FOR ANALOG INPUT	
70	JP 06284 798 A	<input checked="" type="checkbox"/>	SECONDARY EXCITER FOR AC EXCITED SYNCHRONOUS MACHINE	
71	JP 06101 670 A	<input checked="" type="checkbox"/>	ROTOR MOLDING METHOD FOR UNLUBRICATED SCREW COMPRESSOR	
72	JP 06028 319 A	<input checked="" type="checkbox"/>	LOGICAL SIMULATOR	
73	JP 06028 187 A	<input checked="" type="checkbox"/>	DATA LOADING METHOD AND ARITHMETIC PROCESSOR USING THE SAME	
74	JP 06019 708 A	<input checked="" type="checkbox"/>	INSTRUCTION STRING SWITCHING METHOD AND ARITHMETIC PROCESSOR USING THE METHOD	
75	JP 05345 450 A	<input checked="" type="checkbox"/>	RECORDING DEVICE	
76	JP 05292 736 A	<input checked="" type="checkbox"/>	DC POWER SOURCE	
77	JP 05273 813 A	<input checked="" type="checkbox"/>	IMAGE FORMING DEVICE	
78	JP 05260 796 A	<input checked="" type="checkbox"/>	FIELD CONTROL METHOD FOR SYNCHRONOUS MOTOR	
79	JP 05176 258 A	<input checked="" type="checkbox"/>	POWER SUPPLY EQUIPMENT FOR TELEVISION RECEIVER	
80	JP 05104 943 A	<input checked="" type="checkbox"/>	CONTROLLER OF VEHICLE	
81	JP 05088 667 A	<input checked="" type="checkbox"/>	OPERATION CONTROLLER	
82	JP 05033 608 A	<input checked="" type="checkbox"/>	METHOD AND DEVICE FOR CONTROLLING POWER RECOVERY DEVICE	
83	JP 04191 204 A	<input checked="" type="checkbox"/>	AUTOMATIC WAREHOUSE	



US006047115A

United States Patent [11] Patent Number: 6,047,115 Date of Patent: Apr. 4, 2000

Mohan et al.

[54] METHOD FOR CONFIGURING FPGA MEMORY PLANES FOR VIRTUAL HARDWARE COMPUTATION

[75] Inventors: Sundararaj Mohan, Cupertino; Stephen M. Trimberger, San Jose, both of Calif.

[73] Assignee: Xilinx, Inc., San Jose, Calif.

[21] Appl. No.: 08/865,386

[22] Filed: May 29, 1997

[51] Int. Cl.⁷ G06F 17/00

[52] U.S. Cl. 395/500.17; 395/500.18; 712/22; 712/10; 712/11; 712/14; 712/15

[58] Field of Search 364/490; 395/500.17; 500.18; 712/22, 10, 11, 14, 15

[56] References Cited

U.S. PATENT DOCUMENTS

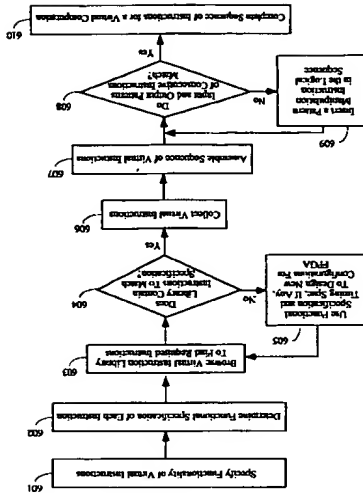
3,775,756 11/1973 Balser 340/172.5
4,237,545 12/1980 Deglin et al. 364/900
4,288,658 9/1981 Bieber et al. 178/3

FOREIGN PATENT DOCUMENTS

53-121537 10/1978 Japan

OTHER PUBLICATIONS

Trimberger et al "A Time-Multiplexed FPGA," IEEE, pp. 22-28, Jan. 1997,
Li et al "Routability Improvement Using Dynamic Intercon-
nect Architecture," IEEE, pp. 61-67, Jan. 1995.
"The Programmable Logic Data Book" copyright 1994,
Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124.
"Principles of CMOS VLSI Design, A Systems Perspec-
tive", N. Weste & K. Eshraghian, Addison-Wesley Publish-
ing Company, 1988, pp. 160-164.



9 Claims, 6 Drawing Sheets

ABSTRACT

A dynamically reconfigurable FPGA includes an array of tiles on a logic plane and a plurality of memory planes. Each tile has associated storage elements on each memory plane, large amounts of data to pass from one FPGA configuration (memory plane) to another with no external memory access, thereby transferring data to/from the storage elements in the logic plane at very high speed. Typically, all the local memory can be simultaneously transferred to/from other memory planes in one cycle. Each FPGA configuration provides a virtual instruction. The present invention uses two different types of virtual instructions: computational and pattern manipulation instructions. Computational instructions perform some computation with data stored in some pre-defined local memory pattern. Pattern manipulation instructions move the local data into different memory locations to create the pattern required by the next instruction. A virtual computation may be accomplished by a sequence of instructions.

Primary Examiner—Kevin J. Teska
Assistant Examiner—Vulke Siek
Attorney, Agent, or Firm—Adam H. Tachner, Jeanette S. Harms

Paper presented at IEEE Workshop on FPGA's for Custom Computing Machines, FCCM '93, Apr. 1993 entitled "Virtual Wires: Overcoming Pin Limitations in FPGA-based Logic Emulators", Jonathan Babb, et al. Jan. 26, 1993, pp. 1-15.
"The Programmable Logic Data Book" copyright 1996, published by Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124, pp 4-291 to 4-302.
David M. Lewis, "Hierarchical Compiled Event-Driven Logic Simulation", Department of Electrical Engineering, University of Toronto, IEEE, 1989, pp. 498-501.
Trimberger, Stephen M., "A Reconfigurable Gate Array and Applications," Proceedings of the IEEE, vol. 81, No. 7, Jul. 1993, pp. 1030-1041.

(List continued on next page.)

	Docum ent ID	U	Title	Current OR
84	JP 04133 682 A	<input checked="" type="checkbox"/>	POWER CONTROL CIRCUIT	
85	JP 04030 910 A	<input checked="" type="checkbox"/>	NUMERICALLY CONTROLLED DEVICE PROVIDED WITH SYNCHRONOUS TAPPING FUNCTION	
86	JP 04018 657 A	<input checked="" type="checkbox"/>	MICROCOMPUTER I/O BUS	
87	JP 03290 710 A	<input checked="" type="checkbox"/>	INDUSTRIAL ROBOT CONTROLLER	
88	JP 03212 191 A	<input checked="" type="checkbox"/>	CONTROLLER FOR CYCLOCONVERTER	
89	JP 03154 514 A	<input checked="" type="checkbox"/>	SEMICONDUCTOR INTEGRATED CIRCUIT	
90	JP 03119 588 A	<input checked="" type="checkbox"/>	BLOCK SYNCHRONOUS COUNTER CIRCUIT	
91	JP 03078 254 A	<input checked="" type="checkbox"/>	P-TYPE MOS TRANSISTOR	
92	JP 03052 566 A	<input checked="" type="checkbox"/>	POWER SUPPLY CONTROLLING CIRCUIT OF INVERTOR	
93	JP 03036 602 A	<input checked="" type="checkbox"/>	DUAL CONTROL SYSTEM	
94	JP 03026 804 A	<input checked="" type="checkbox"/>	STEAM TURBINE CONTROLLER	
95	JP 02277 938 A	<input checked="" type="checkbox"/>	FUEL FEED CONTROL DEVICE OF INTERNAL COMBUSTION ENGINE	
96	JP 02241 936 A	<input checked="" type="checkbox"/>	ENGINE CONTROLLER FOR VEHICLE WITH AUTOMATIC TRANSMISSION	
97	JP 02228 252 A	<input checked="" type="checkbox"/>	PWM DEVICE	
98	JP 02193 593 A	<input checked="" type="checkbox"/>	OPERATING METHOD FOR SYNCHRONOUS MOTOR	
99	JP 02086 234 A	<input checked="" type="checkbox"/>	FRAME SYNCHRONIZING CIRCUIT	
100	JP 02048 396 A	<input checked="" type="checkbox"/>	BELT HOIST	
101	JP 02041 694 A	<input checked="" type="checkbox"/>	SYNCHRONOUS OPERATION CONTROLLER FOR MAIN SHAFT	
102	JP 02041 693 A	<input checked="" type="checkbox"/>	SYNCHRONOUS OPERATION CONTROLLER FOR MAIN SHAFT	
103	JP 01311 829 A	<input checked="" type="checkbox"/>	OPERATION CONTROLLER FOR GENERATING EQUIPMENT	
104	JP 01273 854 A	<input checked="" type="checkbox"/>	CONTROLLER FOR ENGINE	
105	JP 01219 593 A	<input checked="" type="checkbox"/>	PROTECTIVE BARRIER OF NUCLEAR FUSION DEVICE	
106	JP 01206 890 A	<input checked="" type="checkbox"/>	CRANE OPERATION CONTROLLER	

Laung-Terng Wang, et al. "SSIM: A Software Levelized Compiled-Code Simulator", 24th ACM/IEEE Design Automation Conference, 1987, Paper 2.1, pp 2-8.

Randal E. Bryant, et al. "COSMOS: A Compiled Simulator for MOS Circuits", 24th ACM/IEEE Design Automation Conference, 1987, Paper 2.2, pp 9-16.

Peter M. Maurer, "Scheduling Blocks of Hierarchical Compiled Simulation of Combinational Circuits", IEEE Transactions on Computer-Aided Design, vol. 10, No. 2, Feb. 1991, pp 184-192.

Narasimha B. Bhat, Kamal Chaudhary, and Ernest S. Kuh, "Performance-Oriented Fully Routable Dynamic Architecture for a Field Programmable Logic Device", Electronic Research Laboratory, College of Engineering, University of California, Berkeley, Jun. 1, 1993.

Andre DeHon, "DPGA-Coupled Microprocessors: Commodity ICs for the Early 21st Century", NE43-791, 545 Technology Square, Cambridge, MA 02139, 10 pages, Jan. 6, 1994.

Chi-Yuan Chin, et al., "A Dynamically Reconfigurable Interconnection Chip" Session XX: Special Purpose Accelerators, IEEE International Solid State Circuits Conference, pp 276-177, 425, Feb. 27, 1987.

Trimberger, Stephen M. and Chene, Mon-Ren, "Placement-Based Partitioning for Lookup-Table-Based FPGAs," IEEE, Computer Design-ICCD '92, 1992 International Conference, pp. 91-94.

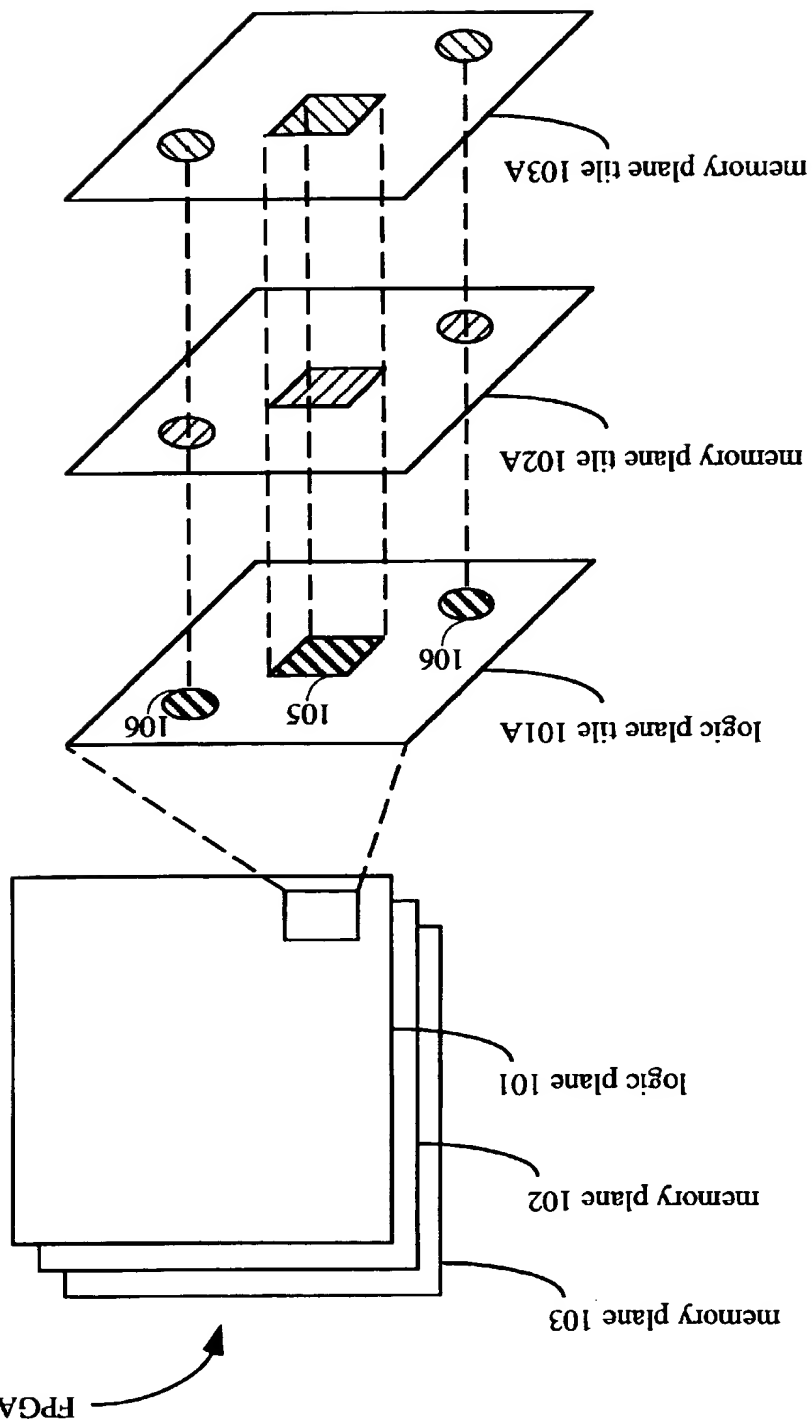
U.S. PATENT DOCUMENTS

OTHER PUBLICATIONS

4,434,461	2/1984	Publ
4,750,155	6/1988	Hsieh
4,821,233	4/1989	Hsieh
5,019,996	5/1991	Lee
5,142,242	9/1992	Zelenga et al.
5,224,056	6/1993	Chene et al.
5,349,248	9/1994	Parlour et al.
5,377,331	12/1994	Dierup et al.
5,426,378	6/1995	Ong
5,426,738	6/1995	Hsieh et al.
5,426,744	6/1995	Sawase et al.
5,432,388	7/1995	Crafts et al.
5,457,644	10/1995	McCollum
5,469,577	11/1995	Eag et al.
5,513,124	4/1996	Trimberger et al.
5,521,835	5/1996	Trimberger et al.
5,526,276	6/1996	Cox et al.
5,530,654	6/1996	Takahashi
5,583,450	12/1996	Trimberger et al.
5,600,263	2/1997	Trimberger et al.
5,629,637	5/1997	Trimberger et al.
5,646,545	7/1997	Trimberger et al.
5,784,313	7/1998	Trimberger et al.
5,802,290	9/1998	Casselman
364/200		
365/203		
365/205		
364/483		
326/38		
364/490		
326/41		
326/39		
326/93		
326/38		
365/182		
395/200.31		

	Docum ent ID	U	Title	Current OR
107	JP 01131 867 A	<input checked="" type="checkbox"/>	ICE-MAKING DEVICE OF REFRIGERATOR OR THE LIKE	
108	JP 01100 696 A	<input checked="" type="checkbox"/>	HOME CONTROLLER	
109	JP 01092 550 A	<input checked="" type="checkbox"/>	AIR-FUEL INJECTION CONTROL AMOUNT DEVICE FOR INTERNAL COMBUSTION ENGINE	
110	JP 01078 127 A	<input checked="" type="checkbox"/>	SHIFT FEELING EVALUATING METHOD	
111	JP 01004 155 A	<input checked="" type="checkbox"/>	CHANNEL SYNCHRONIZING CONTROL SYSTEM	
112	JP 63314 351 A	<input checked="" type="checkbox"/>	ELECTRONIC CONTROL FUEL INJECTOR OF INTERNAL COMBUSTION ENGINE	
113	JP 63253 878 A	<input checked="" type="checkbox"/>	SWITCHING POWER-SUPPLY DEVICE	
114	JP 63235 734 A	<input checked="" type="checkbox"/>	ROLLER SYNCHRONIZING TYPE ONE-WAY CLUTCH	
115	JP 63228 970 A	<input checked="" type="checkbox"/>	METHOD OF CONTROLLING CURRENT TYPE INVERTER	
116	JP 63225 139 A	<input checked="" type="checkbox"/>	METHOD OF EVALUATING FEELING OF SHIFT	
117	JP 63225 138 A	<input checked="" type="checkbox"/>	METHOD OF EVALUATING FEELING OF SHIFT	
118	JP 63152 729 A	<input checked="" type="checkbox"/>	SYNCHROMESH DEVICE	
119	JP 63152 728 A	<input checked="" type="checkbox"/>	SYNCHROMESH DEVICE	
120	JP 63016 028 A	<input checked="" type="checkbox"/>	GAS DRYER	
121	JP 62233 837 A	<input checked="" type="checkbox"/>	MICROPROGRAM CONTROLLER	
122	JP 62195 426 A	<input checked="" type="checkbox"/>	FUEL SUPPLY CONTROL DEVICE OF ENGINE	
123	JP 62128 236 A	<input checked="" type="checkbox"/>	RECEPTION FRAME SYNCHRONIZING CONTROL SYSTEM	
124	JP 62101 838 A	<input checked="" type="checkbox"/>	CONTROL DEVICE FOR AUXILIARY MACHINE OF ENGINE	
125	JP 62048 267 A	<input checked="" type="checkbox"/>	SYNCHRONOUS CHANGEOVER SYSTEM FOR INVERTER	
126	JP 62023 212 A	<input checked="" type="checkbox"/>	SYNCHRONIZING DETECTION TYPE AMPLITUDE DETECTION CIRCUIT	
127	JP 62006 455 A	<input checked="" type="checkbox"/>	HELICAL SCAN WORKING DEVICE FOR LOWER CYLINDER OF ROTARY HEAD TYPE MAGNETIC RECORDING AND REPRODUCING DEVICE	
128	JP 61193 579 A	<input checked="" type="checkbox"/>	PICTURE ELEMENT POSITION RECOGNITION OF SOLID-STATE IMAGE PICK-UP DEVICE	
129	JP 61144 932 A	<input checked="" type="checkbox"/>	DECODING CIRCUIT	

FIG. 1



	Docum ent ID	U	Title	Current OR
130	JP 61116 996 A	<input checked="" type="checkbox"/>	CONTROLLER OF SYNCHRONOUS MOTOR	
131	JP 61109 499 A	<input checked="" type="checkbox"/>	POWER SYSTEM STABILIZER	
132	JP 61084 441 A	<input checked="" type="checkbox"/>	CONTROL DEVICE IN SYNCHROMESH TYPE SPEED CHANGE GEAR UNIT	
133	JP 61051 526 A	<input checked="" type="checkbox"/>	LOAD CELL BALANCE	
134	JP 61009 185 A	<input checked="" type="checkbox"/>	CONTROLLER OF SYNCHRONOUS MOTOR	
135	JP 60147 575 A	<input checked="" type="checkbox"/>	STARTING OF MULTI-NOZZLE PELTON TURBINE	
136	JP 60055 466 A	<input checked="" type="checkbox"/>	SYNCHRONIZATION SYSTEM BETWEEN PLURAL PROCESSORS	
137	JP 60026 113 A	<input checked="" type="checkbox"/>	ENGINE WITH VARIABLE NUMBER OF OPERATING CYLINDER	
138	JP 60011 754 A	<input checked="" type="checkbox"/>	CONTROLLING METHOD OF AUTOMATIC TRANSMISSION	
139	JP 59136 071 A	<input checked="" type="checkbox"/>	FLICKER COMPENSATING DEVICE	
140	JP 59041 608 A	<input checked="" type="checkbox"/>	RANKINE GENERATOR	
141	JP 58222 797 A	<input checked="" type="checkbox"/>	DEFECT DIAGNOSING DEVICE FOR INVERTER	
142	JP 58205 008 A	<input checked="" type="checkbox"/>	COMBUSTION APPARATUS	
143	JP 58193 398 A	<input checked="" type="checkbox"/>	DEVICE FOR USING SURPLUS POWER OF DC POWER SOURCE USEFUL FOR TREATMENT OF METAL SURFACE OR THE LIKE	
144	JP 58110 811 A	<input checked="" type="checkbox"/>	INTAKE DEVICE OF ENGINE WITH SUPERCHARGER	
145	JP 58107 135 A	<input checked="" type="checkbox"/>	STEAM HEATING TYPE TEA LEAF FINE ROLLING MACHINE	
146	JP 58010 993 A	<input checked="" type="checkbox"/>	CONTROL SYSTEM FOR SYNCHRONIZING MEMORY	
147	JP 57193 803 A	<input checked="" type="checkbox"/>	CONTROLLING DEVICE	
148	JP 57170 100 A	<input checked="" type="checkbox"/>	LOAD CONTROLLING SYSTEM USING SYNCHRONOUS PHASE MODIFIER FOR USE WITH SHAFT GENERATING DEVICE	
149	JP 57142 093 A	<input checked="" type="checkbox"/>	DIGITAL CHROMINANCE SIGNAL GENERATOR	
150	JP 57126 263 A	<input checked="" type="checkbox"/>	WIND POWER GENERATING APPARATUS	
151	JP 57105 519 A	<input checked="" type="checkbox"/>	COMBUSTION CHAMBER FOR INTERNAL COMBUSTION ENGINE	
152	JP 57084 366 A	<input checked="" type="checkbox"/>	APPARATUS FOR INDICATING EFFECTIVE POWER AND REACTIVE POWER OF SYNCHRONOUS MACHINE ON 2-DIMENSIONAL PLANE	

Figure 2

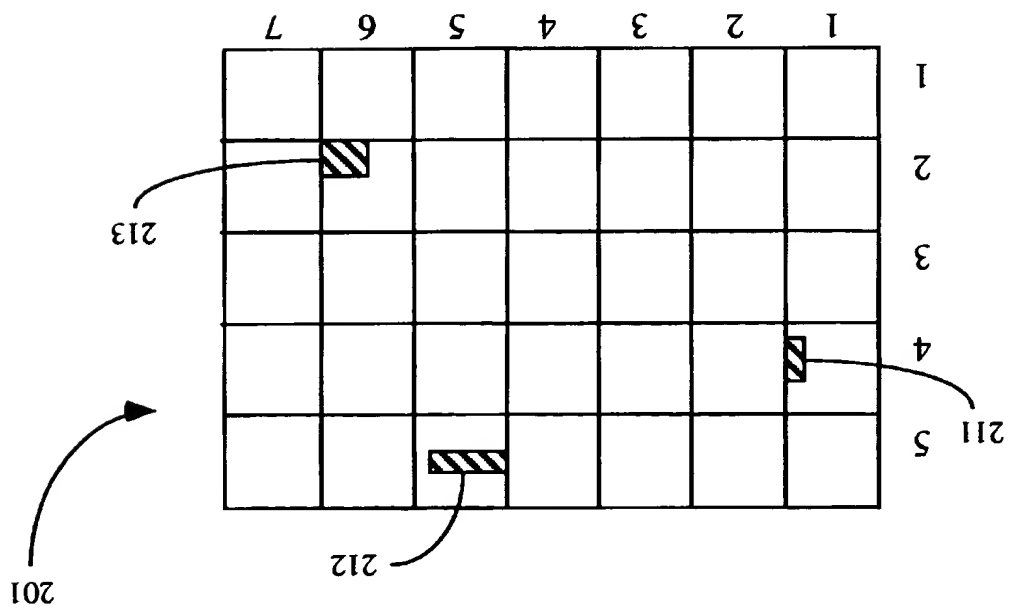
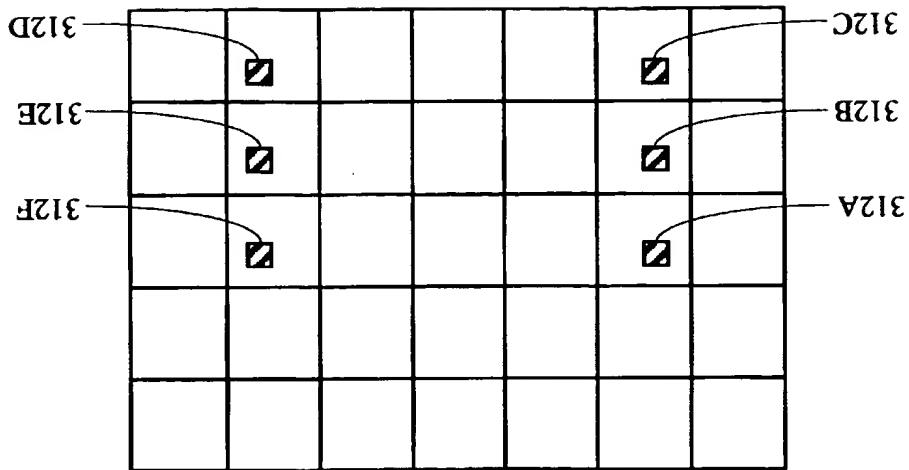


Figure 3C



	Docum ent ID	U	Title	Current OR
153	JP 57052 606 A	<input checked="" type="checkbox"/>	AUTOMATIC CONTROL SYSTEM FOR TURBINE	
154	JP 57032 885 A	<input checked="" type="checkbox"/>	EVACUATING DEVICE OF ELECTRON BEAM WELDING DEVICE	
155	JP 56098 506 A	<input checked="" type="checkbox"/>	METHOD OF FOLLOW-UP CONTROL OF SYNCHRONIZING MEANS FOR CONTROLLING GOVERNOR LOAD RESTRICTOR OF TURBINE	
156	JP 56049 689 A	<input checked="" type="checkbox"/>	SYNCHRONOUS STARTING DEVICE FOR DC MOTOR	
157	JP 55142 977 A	<input checked="" type="checkbox"/>	OPERATION OF WATER WHEEL OR PUMP WATER WHEEL	
158	JP 55109 094 A	<input checked="" type="checkbox"/>	LOAD SHARING SYSTEM	
159	JP 55100 067 A	<input checked="" type="checkbox"/>	CONTROLLING METHOD OF POWER TRANSISTOR	
160	JP 55093 913 A	<input checked="" type="checkbox"/>	TURBINE CONTROL SYSTEM	
161	JP 55074 618 A	<input checked="" type="checkbox"/>	OPERATING FREQUENCY SELECTOR FOR DIGITAL COMPUTER	
162	JP 55060 868 A	<input checked="" type="checkbox"/>	CIRCUIT FOR MEASURING CONTACT RESISTANCE OF CONTACT MEMBER DURING ITS OPENING AND CLOSING	
163	JP 55032 104 A	<input checked="" type="checkbox"/>	METHOD AND APPARATUS FOR SPEED CONTROL IN INTERMITTENT LOAD OPERATION	
164	JP 55018 247 A	<input checked="" type="checkbox"/>	CENTRIFUGAL DEHYDRATOR	
165	JP 54158 502 A	<input checked="" type="checkbox"/>	ROTOR STRESS ESTIMATING TURBINE CONTROLLER	
166	JP 54150 642 A	<input checked="" type="checkbox"/>	ENERGY STORAGE SYSTEM	
167	JP 54087 840 A	<input checked="" type="checkbox"/>	NOISELESS POWER CONTROL SYSTEM	
168	JP 54082 501 A	<input checked="" type="checkbox"/>	ROTOR STRESS FORECASTING TURBINE CONTROL SYSTEM	
169	JP 54076 913 A	<input checked="" type="checkbox"/>	HYSTERESIS MOTOR	
170	JP 53083 019 A	<input checked="" type="checkbox"/>	LOAD SELECTOR CIRCUIT FOR PLURALITY OF INVERTERS	
171	JP 52091 277 A	<input type="checkbox"/>	APPARATUS FOR EQUALIZING OPERATION SPEED	

Figure 3A

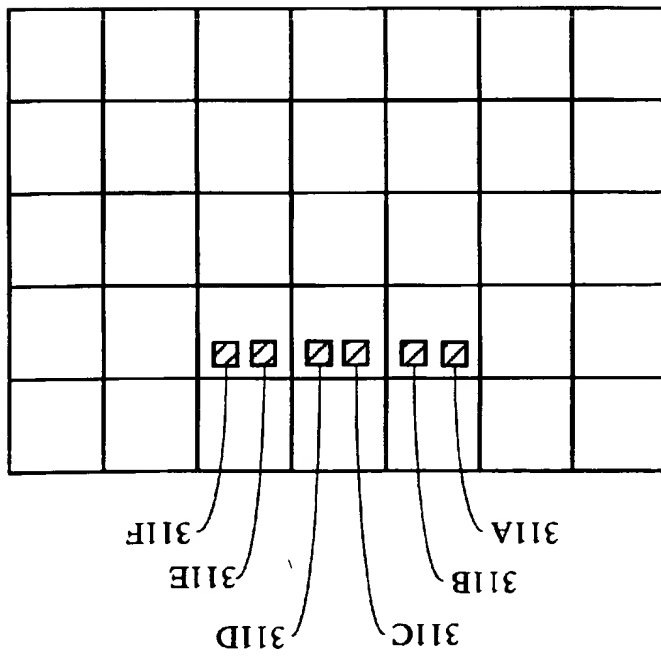


Figure 3B

